

Interner Bericht

DLR-IB-FT-BS-2022-202

**Entwicklung einer Missions-
simulation für solarelektrische
stationierbare Stratosphären-
flugzeuge zur Analyse verschie-
dener Flugzeugkonfigurationen**

Hochschulschrift

Thomas Müller

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik
Braunschweig



DLR

Deutsches Zentrum
für Luft- und Raumfahrt

Institutsbericht
DLR-IB-FT-BS-2022-202

**Entwicklung einer Missionssimulation für solarelektrische
stationierbare Stratosphärenflugzeuge zur Analyse
verschiedener Flugzeug-Konfigurationen**

Thomas Müller

Institut für Flugsystemtechnik
Braunschweig

129 Seiten
103 Abbildungen
31 Tabellen
62 Referenzen

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugsystemtechnik
Abteilung Sichere Systeme & Systems Engineering

**Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird
elektronisch ohne Einschränkungen in ELIB abgelegt.**

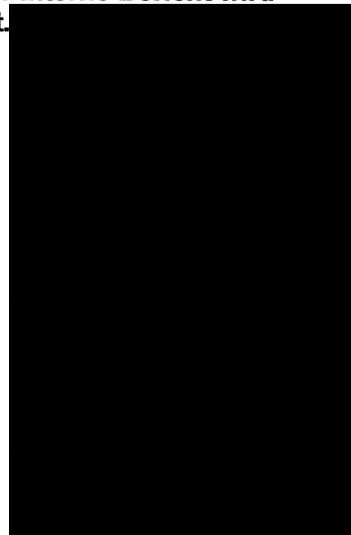
Braunschweig, den 31.01.2022

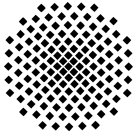
Institutsleitung: Prof. Dr.-Ing. S. Levedag

Abteilungsleitung: Andreas Bierig

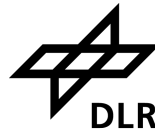
Betreuer:in: Andreas Bierig

Verfasser:in: Thomas Müller





Universität Stuttgart
Institut für Flugzeugbau



DLR Braunschweig
Institut für Flugsystemtechnik

Masterthesis

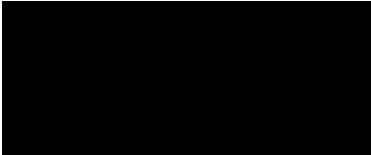
Entwicklung einer Missionssimulation für solarelektrische stationierbare Stratosphärenflugzeuge zur Analyse verschiedener Flugzeug-Konfigurationen

Development of a mission simulation for stationable solar-electric stratospheric aircraft to analyse different aircraft configurations

Autor: M.Sc. Thomas Müller

Betreuung:
Universität Stuttgart Prof. Dr.-Ing. Andreas Strohmeyer
Universität Stuttgart M.Sc. Johannes Schneider
DLR Braunschweig Dipl.-Ing. Andreas Bierig

Eingereicht: im Januar 2022



Masterarbeit

Thema:

Entwicklung einer Missionssimulation für solarelektrische stationierbare Stratosphärenflugzeuge zur Analyse verschiedener Flugzeug-Konfigurationen

Hintergrund

Das Deutsche Zentrum für Luft- und Raumfahrt forscht an innovativen Lösungen für die Umsetzung solargetriebener Flugzeuge für den dauerhaften Betrieb in Einsatzhöhen von 20 km. Dabei stehen insbesondere Ansätze im Fokus, die die Kosten für derartige Luftfahrzeuge deutlich senken und die Robustheit andererseits weiter erhöhen werden.

Aufgabenstellung

Im Rahmen Ihrer Abschlussarbeit wirken Sie am Aufbau einer Auslegungs- und Simulationsumgebung für solarelektrische Höhenflugzeuge mit dem Ziel schnell und aussagesicher die Auswirkungen neuartiger Technologien auf Konfigurationen sowie auf die Flugleistungen bewerten zu können. Hierbei bearbeiten Sie insbesondere Fragestellungen im Bereich des Konfigurationsentwurfes, des elektrischen Antriebsstrangs, der Thermalsimulation oder grundsätzliche Themen aus dem Bereich der Simulationstechnik oder der Entwurfsoptimierung. Ebenfalls sind experimentelle Arbeiten unter Nutzung unserer einzigartigen Versuchsstände durchführbar.

Arbeitspunkte:

- Literaturrecherche
- Gliederung und Identifikation der missionsrelevanten Parameter
- Aufbau einer Missionssimulation basierend auf Matlab
- Verifikation der Simulation anhand vorhandener Solarflugzeuge (z.B. Helios, Zephyr)
- Analysen zur Mission zum verfügbaren Projektstand HAP
- Dokumentation der Arbeit
- Präsentation der Arbeit

Arbeit ausgegeben am: 01.07.2021

Arbeit abgegeben am:

Betreuer: Johannes Schneider (IFB),

Andreas Bierig (DLR)

Stuttgart, 17.05.2021

Prof. Dr.-Ing. Andreas Strohmayer

Kurzfassung

Im fortschreitenden Verlauf einer Flugzeugentwicklung resultieren üblicherweise Abweichungen in den zuvor projektierten Kenngrößen verschiedenster Komponenten. Dazu zählen Bauteilgewichte, sowie Energiedichten und Wirkungsgrade, welche wiederum Einfluss auf die Flugleistungen nehmen. Zur Absicherung der angepeilten Flugleistungsgrößen kann noch vor dem Erstflug bzw. meist vor dem Bau größerer Baugruppen eine virtuelle Simulation durchgeführt werden. Die simulierten Missionsdaten und -ergebnisse können zukünftig genutzt werden zur vorzeitigen Einflussnahme kritischer Komponenten sowie zur Vermeidung ungeplanter Mehrkosten.

Im Rahmen dieser Arbeit soll ein Missionssimulator für das solarelektrische Stratosphärenflugzeug High Altitude Platform (HAP) entstehen. Die Umsetzung erfolgt ausschließlich in Matlab in objektorientierter Programmierung. Dabei soll die Objektorientierung durch ihren Bezug zum Simulationsprogramm einen einfachen Einstieg für Personen ermöglichen, die erstmals mit dem Programm arbeiten um eventuelle Änderungen und Erweiterungen zu erleichtern und zu beschleunigen.

Beginnend mit dem Überblick über das Projekt HAP und den Kenngrößen des Flugzeuges wird darauf aufbauend die Klassenstruktur der Objekte erarbeitet. Diese wird soweit möglich an die Komponenten angelehnt, um die Handhabung intuitiv zu gestalten. Anschließend folgt die Ausarbeitung der Objekte und ihrer Funktionen, die auf Formeln aus betreffender Literatur gestützt sind. Um die simulierten Daten nachzuvollziehen, werden diese Daten gespeichert und können visuell ausgegeben werden. Die Analyse dieser Daten und die Variation einzelner Parameter soll kritische Komponenten und Stellgrößen aufzeigen. Abschließend werden in einem Ausblick mögliche Erweiterungen für die Simulation gegeben und Ansätze zur Zusammenführung der Simulation mit dem Erstflug eines Demonstrators.

Abstract

Deviations in the previously projected parameters of various components often result in the progress of an aircraft development. This leads to component weights, energy densities and efficiencies, which in turn influence the flight performance. To validate the targeted flight performance parameters, a virtual simulation can be performed before the first flight or, in most cases, before the assembling of larger modules. The simulated mission data and results can be used to influence critical components ahead of time, as well as to avoid unplanned additional costs.

Within the scope of this work, a mission simulator for the solar-electric stratospheric aircraft HAP is to be developed. The implementation is done exclusively in Matlab in object-oriented programming. Thereby, it shall provide an easy entry for persons working with the program for the first time due to its relation to the simulation program in order to facilitate and accelerate possible changes and extensions.

Beginning with the overview of the project HAP and the characteristics of the aircraft, a class diagram of the objects is compiled based on them. As far as possible this is based on the components in order to arrange the handling intuitively. This is followed by the elaboration of the objects and their functions, which are based on formulas from relevant literature. In order to reproduce the simulated data, all data are tracked and can be output visually. The analysis of these data and the variation of single parameters shall show critical components and regulating variables. Finally, extension for the simulation will be explained in a scope and approaches to merge the simulation with the first flight of a demonstrator.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Ziel der Arbeit	2
1.3	Gliederung der Arbeit	2
2	Stand der Technik	3
2.1	Hochfliegende Solarplattformen	3
2.1.1	HAP	3
2.1.2	BAE PHASA 35	3
2.1.3	Zephyr S	4
2.1.4	Anforderungen Solarflugzeuge	4
2.2	Technologiestand	5
2.3	Bedarf der Simulation	7
2.4	Vorhandene Simulationssysteme	8
2.4.1	Pacelab APD	9
2.4.2	Aerospace Blockset	10
3	Struktur und Aufbau der Simulation	11
3.1	Grundlagen objektorientierte Programmierung	11
3.2	Aufbau der Programmstruktur	12
3.3	Ablauf der Simulation	13
3.4	Klassen und Funktionen	15
3.4.1	Top Level	15
3.4.1.1	Scenario	15
3.4.1.2	SimLog	17
3.4.2	Mid Level	17
3.4.2.1	Aircraft	17
3.4.2.2	Mission	20
3.4.2.3	Environment	21
3.4.3	Sub Level	22
3.4.3.1	Winglike	22
3.4.3.2	DragProducer	22
3.4.3.3	Battery	23
3.4.3.4	SolarGenerator	24
3.4.3.5	Propulsion	24
3.4.3.6	Propeller	26
3.4.3.7	Motor	26
3.4.3.8	Atmosphere	27
3.4.3.9	Sun	27
3.4.3.10	Wind	28
4	Modellierung des Flugzeuges HAP und seiner Umgebung	29
4.1	Umgebung	29
4.1.1	Atmosphäre	29
4.1.2	Sonnenmodell	30
4.1.3	Windmodell	36
4.2	Flugzeug	38
4.2.1	Koordinatensysteme	38

4.2.2	Fluggeschwindigkeit	40
4.2.3	Flugabschnitte	40
4.2.4	Flügel und Leitwerke	42
4.2.5	Trimmrechnung	46
4.2.6	Widerstände	48
4.2.7	Antriebsstrang	49
4.2.8	Solargenerator	54
4.2.9	Batteriemodell	57
5	Verifikation der Simulation	60
5.1	Methoden zur Verifizierung	60
5.1.1	Flugabschnitte	60
5.1.2	Solarmodell	61
5.1.3	Iterlogs	63
5.2	Verifikation durch HALE Flugzeuge	63
6	Simulationsergebnisse zum Projekt HAP	65
6.1	Stellgrößen der Simulation	65
6.2	Standardmission	68
6.2.1	Eingabeparameter	68
6.2.2	Analyse der Ergebnisse	69
6.2.3	Missionsoptimierung	75
6.3	Erweiterung für Parameterstudien	78
6.3.1	Einschub Studienklasse und -funktionen	78
6.3.2	Potential Studien	82
6.3.3	Beispielstudien	82
7	Zusammenfassung und Ausblick	85
7.1	Fazit	85
7.2	Ausblick	86
A	Verwendete Parameter und Daten	I
A.1	Flugzeugparameter HAP	I
A.2	Umgebungsbezogene Daten	V
A.2.1	Sonnenmodell	V
A.2.2	Winddaten	VII
A.3	Ergänzende Formeln	VII
A.4	Ergänzende Daten	VIII
A.4.1	BAE PHASA 35	VIII
A.4.2	Zephyr S	IX
A.4.3	Daten für Entwurfsdiagramm	X
B	HALE Flugzeuge	XII
B.1	Aurora Flight Sciences Odyssey	XII
B.2	AeroVironment Sunlider	XII
C	Projektverlauf HAP	XIV

D Programmarchiv	XV
D.1 Parameterliste des Programms	XV
D.2 Analyse Stationierbarkeit über Nacht	XX
D.3 Analyse bestes Gleiten	XXI
D.4 Analyse Sinkflug über Nacht	XXII
D.5 Berechnung der Luftdichte	XXIII

Abbildungsverzeichnis

1.1	Erstes Entwurfsmodell HAP	1
2.1	Digitales Entwurfsmodell PHASA-35	3
2.2	Das Solarflugzeug Zephyr S beim Start	4
2.3	Entwurfspunkte für HALE Flugzeuge	5
2.4	Entwurfspunkte für verschiedene Flugzeuge	6
2.5	Aktuelle Batteriedichten 2021	7
2.6	Dauerhafte Stationierung abhängig von der Flächenbelastung	8
2.7	Dauerhafte Stationierung abhängig von der Energiedichte der Batterie	8
2.8	Kosten einer Änderung im Projektfortschritt	9
2.9	Eingabemaske Pacelab Mission Suite	9
2.10	Reichweitenanalyse eines Hybridflugzeuges	10
3.1	Beispieldarstellung einer Klasse	12
3.2	Gesamte Klassenstruktur	13
3.3	Instanzierte Objekte und ihre Struktur	14
3.4	Simulationsablauf	15
3.5	Klasse Scenario inklusive Bezeichnungen	16
3.6	Klasse SimLog inklusive Bezeichnungen	17
3.7	Klasse Aircraft inklusive Bezeichnungen	18
3.8	Ablauf und Aufrufe innerhalb der Funktion CalculateFlighState	19
3.9	Klasse Mission inklusive Bezeichnungen	20
3.10	Klasse Environment	21
3.11	Klasse Winglike	22
3.12	Klasse DragProducer	23
3.13	Klasse Battery	23
3.14	Klasse SolarGenerator	24
3.15	Klasse Propulsion	25
3.16	Ablauf und Aufrufe innerhalb der Funktion GetThrustPower	25
3.17	Klasse Propeller	26
3.18	Klasse Motor	26
3.19	Klasse Atmosphere	27
3.20	Klasse Sun	28
3.21	Klasse Wind	28
4.1	Atmosphärendaten verteilt über die Höhe	29
4.2	Arten der solaren Strahlung	31
4.3	Zenitwinkel der Sonne in der Ebene	33
4.4	Azimut- und Zenitwinkel im geodätischen Koordinatensystem	33
4.5	Deklination der Sonne im Jahresverlauf	34
4.6	Drehwinkel vom erdfesten zum geodätischen KOS	35
4.7	Azimutkorrektur abhängig vom Quadranten	36
4.8	Windgeschwindigkeiten über Hamburg	37
4.9	Windgeschwindigkeiten über Kiruna	37
4.10	Windgeschwindigkeiten über Yuma	38
4.11	Kräfte am Flugzeug im Flug	40
4.12	Profilpolare des Flügels für MH139R	43
4.13	Auftriebskoeffizient über Anstellwinkel für MH139R	45
4.14	Momentenbeiwert über Anstellwinkel für MH139R	45
4.15	Momentengleichgewicht um den Schwerpunkt CoG	46

4.16	Propellerkennlinie über den Fortschrittgrad J	49
4.17	Schubverlauf über den Fortschrittgrad J	51
4.18	Drehzahl des Propellers über dem Drehmoment abhängig von der Flughöhe	51
4.19	Quadranten und Vorzeichen der Motorkennlinie	52
4.20	Motorkennlinie des Flugzeuges HAP	53
4.21	Betriebspunkt des Motors abhängig von der Flughöhe	54
4.22	Winkel der Solarpanele am Flugzeug	56
4.23	Max. Batteriekapazität abhängig von der Batterietemperatur	58
4.24	Spezifische Heizleistung abhängig von Tageszeit und Höhe	59
5.1	Unittest für Aircraft.CalculateFlightState	61
5.2	Solare Strahlung ε_{dir} in Abhängigkeit des Höhenwinkel α_S	61
5.3	Sonnenverlauf im Winter für $\phi = 48^\circ$ N; $h = 20.000$ m	62
5.4	Azimutverlauf im Winter für $\phi = 48^\circ$ N; $h = 20.000$ m	62
5.5	Iterlog Parameter verschiedener Objektfunktionen	63
6.1	Einfluss des Parameters iter.dt auf den Missionsverlauf	66
6.2	Einfluss des Parameters iter.da auf den Missionsverlauf	66
6.3	Einfluss des Parameters iter.threshold auf den Missionsverlauf	67
6.4	Einfluss der Iterationsmethoden auf die Simulationszeit	67
6.5	Flugpfad im geodätischen KOS	69
6.6	Verlauf der Höhe über die Missionszeit	70
6.7	Verlauf des Batterieladestands über die Missionszeit	70
6.8	Verlauf der ges. elektr. Leistung über die Missionszeit	71
6.9	Verlauf der Solar Generator Leistung über die Missionszeit	71
6.10	Leistungsparameter eines Motors über die Missionszeit	72
6.11	Verlauf der Betriebspunkte im Motorkennfeld	72
6.12	Heizleistung der Batterien über die Missionszeit	73
6.13	Solare Strahlung ε_{dir} über die Missionszeit	73
6.14	Bahnneigungswinkel γ über die Missionszeit	74
6.15	Widerstandskoeffizient c_D über die Missionszeit	74
6.16	Höhenverlauf abhängig vom Propellerdurchmesser über die Missionszeit	75
6.17	SoC abhängig vom Propellerdurchmesser über die Missionszeit	76
6.18	Motordrehmoment abhängig vom Propellerdurchmesser über die Missionszeit	76
6.19	Motordrehzahl abhängig vom Propellerdurchmesser über die Missionszeit	77
6.20	Höhenwinkel α_S für $\phi = 32^\circ$ N, $n = 142$, $h = 20.000$ m	77
6.21	Verlauf des SoC abhängig von der Startuhrzeit t_{start}	78
6.22	Gesamte Klassenstruktur für Studien	79
6.23	Klasse Study inklusive Bezeichnungen	80
6.24	Ablauf der Studie	81
6.25	Min. SoC und elektr. Energie E der Studie über Heiztemperatur T_{heat}	83
6.26	Min. SoC der Studie über Tag im Jahr n	84
A.1	Profilpolare des Pendelleitwerks für NACA0009	IV
A.2	Auftriebskoeffizient über Anstellwinkel für NACA0009	IV
A.3	Momentenbeiwert über Anstellwinkel für NACA0009	V
A.4	Komponenten der Transformationsmatrix T_{ge} im Winter für $\phi = 48^\circ$ N; $h = 20.000$ m	V
A.5	Sonnenverlauf im Sommer für $\phi = 48^\circ$ N; $h = 20.000$ m	VI
A.6	Azimutverlauf im Sommer für $\phi = 48^\circ$ N; $h = 20.000$ m	VI
A.7	Komponenten der Transformationsmatrix T_{ge} im Sommer für $\phi = 48^\circ$ N; $h = 20.000$ m	VII
B.1	Das Solarflugzeug Aurora Odysseus	XII
B.2	Das Flugzeug Helios der NASA	XIII

B.3	Aufnahme des Sunlider in der Stratosphere	XIII
C.1	Zeitlicher Verlauf Projekt HAP	XIV
D.1	Verlauf des SoC im Reiseflug über Nacht für verschiedene Flughöhen	XX
D.2	Höhenverlauf für verschiedene Fluggeschwindigkeiten	XXI
D.3	Verlauf des SoC für verschiedene Fluggeschwindigkeiten	XXI
D.4	Verlauf des SoC für beste Leistungsparameter	XXII
D.5	Verlauf des SoC bei Sinkflug über Nacht	XXII
D.6	Verlauf der Energien bei Sinkflug über Nacht	XXIII
D.7	Berechnung der Luftdichte	XXIV

Tabellenverzeichnis

2.1	Verschiedene Solarzellen mit Kenndaten	6
3.1	Parameter des Arrays iter	16
3.2	Beispieldaten eines Abschnittes	20
3.3	Abschnittsarten	21
4.1	Atmosphärendaten auf Meereshöhe	30
4.2	Annahmen bezüglich Kräftegleichgewicht	41
4.3	Motorkenndaten des Flugzeuges HAP	52
6.1	Gesetzte Parameter des Arrays iter	65
6.2	Schlüsselparameter des Flugzeuges HAP	68
6.3	Zeit- und ortbezogene Daten zur Mission_0	69
6.4	Bahndaten zur Mission_0	69
6.5	Mögliche Ziele für Ein-Parameter-Studien	82
6.6	Mögliche Ziele für Zwei-Parameter-Studien	82
6.7	Wertebereich des Studienparameters T_{heat}	83
6.8	Wertebereich des Studienparameters n	83
A.1	Kenndaten des Flugzeuges HAP *1	I
A.2	Kenndaten des Flugzeuges HAP *2	II
A.3	Geographische Breiten- und Längengrade der Windmodelle	VII
A.4	Kenndaten des Flugzeuges PHASA 35	VIII
A.5	Geometrische Daten zu PHASA 35	VIII
A.6	Berechnete Daten zu PHASA 35	IX
A.7	Kenndaten des Flugzeuges Zephyr S	IX
A.8	Geometrische Daten zu Zephyr S	IX
A.9	Berechnete Daten zu Zephyr S	X
A.10	Kenndaten zu HAP	X
A.11	Kenndaten zu BAE PHASA 35	X
A.12	Kenndaten zu Zephyr S	XI
A.13	Kenndaten zu EGenius	XI
A.14	Kenndaten zu AS33Me	XI
A.15	Kenndaten zu Icare II	XI
C.1	Unterschiede HAP α und HAP Ω	XIV

Symbolverzeichnis

Lateinische Symbole

Symbol	Einheit	Bedeutung
A_{SG}	m^2	Gesamtfläche des Solargenerators
A_{SPi}	m^2	Fläche des Solarpanels i
a_0	m/s	Schallgeschwindigkeit auf Meereshöhe
b_i	m	Spannweite des jeweiligen Flügels
c	m	Referenzlänge des Flügel
c_D	–	Widerstandskoeffizient
c_{D0}	–	Nullwiderstandsbeiwert
c_{Di}	–	Widerstandsbeiwert der Komponente i
c_{Dfus}	–	Widerstandsbeiwert des Rumpfes
c_{D0fus}	–	Nullwiderstandsbeiwert des Rumpfes
c_L	–	Auftriebskoeffizient des Flügels
c_M	–	Momentenbeiwert des Flügels
c_P	–	Leistungskoeffizienten
c_T	–	Schubbeiwert
c_{Wing}	m	Bezugsflügelteufe des Flügels
c_l	–	Auftriebskoeffizient des Flügelprofils
$c_{l=0}$	–	Nullauftriebskoeffizient
c_m	–	Momentenbeiwert des Flügelprofils
$c_{m0,a}$	–	Nullmomentenbeiwert des Flügelprofils außen
$c_{m0,i}$	–	Nullmomentenbeiwert des Flügelprofils innen
$\Delta c_{m0}/\varepsilon_t$	–	Auswirkung einer linearen Verwindung
D	N	Widerstandskraft
D_{Prop}	m	Durchmesser des Propellers
d_{fus}	m	Durchmesser des Rumpfes
dt	s	Zeitabstand eines Simulationsschrittes
E	Ws	elektrische Energie
E_{heat}	Ws	Energie zum Heizen der Batterien
E_i	Ws	Energieladung oder -verbrauch eines Simulationsschrittes
e	–	Oswaldfaktor
e_{bat}	Ws/kg	Energiedichte der Batterien
e_{batT}	Ws/kg	Energiedichte bei Temperatur T
e_{batT25}	Ws/kg	Energiedichte bei Temperatur T = 25°C
e_{bat0}	Ws/kg	Theor. mögliche Energiedichte
FF_{fus}	–	Formfaktor Rumpf
f_0	%	rel. Feuchte auf Meereshöhe
g_0	m/s	Erdbeschleunigung auf Meereshöhe
$g(h)/g$	m/s	Erdbeschleunigung
h	m, km	Flughöhe
h_s	km	Skalenhöhe
h_{target}	m	Zielhöhe
IF_{fus}	–	Interferenzfaktor Rumpf
J	–	Fortschrittgrad
$k_{M Mot}$	$1/Nms$	Drehmomentkonstante
L	N	Auftriebskraft

L_{HTail}	N	Auftrieb des Höhenleitwerks
L_{req}	N	benötigter Auftrieb am Flugzeug
L_{Wing}	N	Auftrieb des Flügels
l_{fus}	m	Länge des Rumpfes
M_{HTail}	Nm	Moment des Höhenleitwerks
M_{Mot}	Nm	Drehmoment des Motors
$M_{max Mot}$	Nm	max. Drehmoment
$M_{min Mot}$	Nm	min. Drehmoment
M_{Prop}	Nm	Drehmoment des Propellers
M_{Wing}	Nm	Moment des Flügels
m	kg	Flugzeugmasse
m_{bat}	kg	Batteriemasse
m_0	kg	Flugzeugmasse
n	–	Tag im Jahr
n_{chgi}	–	Äquivalenter Ladezyklus i
n_{eng}	–	Anzahl der Antriebe
n_{Mot}	$1/s$	Drehzahl des Motors
n_{Prop}	$1/s$	Drehzahl des Propellers
n_{sim}	–	Nummer der Simulationsschleife
n_0	–	Starttag
$n_0 Mot$	$1/s$	Leerlaufdrehzahl
P	W	Leistung
$P_{el Mot}$	W	elektr. Leistung des Motors
$P_{el tot Mot}$	W	gesamte elektr. Leistung der Antriebe
P_{heat}	W	Heizleistung
$P_{max Mot}$	W	max. Leistung des Motors
$P_{max rcu Mot}$	W	max. Leistung des Motors bei Rekuperation
P_{Prop}	W	aerodynamische Leistung des Propellers
$P_{Prop shaft}$	W	Wellenleistung des Propellers
P_{SG}	W	Energie des Solargenerators
$P_{SP i}$	W	Energie des Solarpanels i
$P_{shaft Mot}$	W	Leistung der Abtriebswelle
p_0	hPa	Luftdruck auf Meereshöhe
p_{heat}	W/kg	spezifische Heizleistung
p_{sun}	W/m^2	spezifische Sonnenstrahlung im geodätischen KOS
$p_{sun i}$	W/m^2	spezifische Sonnenstrahlung des individuellen Panels
$(p_{sun})_f$	W/m^2	spezifische Sonnenstrahlung im flugzeugfesten KOS
$(p_{sun})_{pi}$	W/m^2	spezifische Sonnenstrahlung im individuellen Panel KOS
q	N/m^2	Staudruck
Re	–	Reynoldszahl
R_0	m	mittlerer Erdradius
r_{turn}	m	Kurvenradius des geflogenen Kreises
S	m^2	Flügelfläche
S_i	m^2	Flügelfläche des jeweiligen Flügels
SoC	–	State of Charge
SoC_{min}	–	minimaler State of Charge
SoC_0	–	State of Charge bei Missionsbeginn
S_{Wing}	m^2	Flügelfläche des Flügels
$S_{wet i}$	m^2	bespülte Oberfläche der Komponente i

T	N	Schubkraft
T_i	N	Schub eines Antriebes
T_{heat}	$^{\circ}C$	Heiztemperatur Batterien
T_{Prop}	N	Schub des Propellers
T_{fg}	–	Drehmatrix von geodätisch nach flugzeugfest
T_{ge}	–	Drehmatrix von erdfest nach geodätisch
T_{req}	N	erforderliche Gesamtschubkraft
$T_{x\phi}$	–	Drehmatrix um die X-Achse mit dem Winkel ϕ
$T_{x\phi_{roll}}$	–	Drehmatrix um die X-Achse mit dem Winkel ϕ_{roll}
$T_{x\phi_{SP}}$	–	Drehmatrix um die X-Achse mit dem Winkel ϕ_{SP}
$T_{y\alpha_S}$	–	Drehmatrix um die Y-Achse mit dem Winkel α_S
$T_{y\theta}$	–	Drehmatrix um die Y-Achse mit dem Winkel θ
$T_{y\theta_{SP}}$	–	Drehmatrix um die Y-Achse mit dem Winkel θ_{SP}
$T_{z\Psi}$	–	Drehmatrix um die Z-Achse mit dem Winkel Ψ
$T_{z\Psi_{delta}}$	–	Drehmatrix um die Z-Achse mit dem Winkel Ψ_{delta}
T_0	K	Temperatur auf Meereshöhe
T_{1z}	–	Drehmatrix um die Z-Achse
T_{2y}	–	Drehmatrix um die Y-Achse
T_{3z}	–	Drehmatrix um die Z-Achse
T_{4y}	–	Drehmatrix um die Y-Achse
t	h	lokale Uhrzeit
$t_{Missionsende}$	s	Zeitpunkt des Missionsendes
t_{target}	h	Zielzeit
t_{Start}	h	Startuhrzeit
t_0	h	Startuhrzeit
v	m/s	Geschwindigkeit
\dot{v}	m/s^2	Beschleunigung
v_{EAS}	m/s	Fluggeschwindigkeit in EAS
v_{TAS}	m/s	Fluggeschwindigkeit in TAS
v_w	m/s	Windgeschwindigkeit
$x_{ACHTail}$	m	Abstand des Neutralpunktes des Höhenleitwerks zur Nase
x_{ACWing}	m	Abstand des Neutralpunktes des Fluges zur Nase
x_{CoG}	m	Abstand des Schwerpunkts zur Nase
x_E	–	X-Achse des erdfesten Koordinatensystems
x_a	–	X-Achse des aerodynamischen Koordinatensystems
x_f	–	X-Achse des flugzeugfesten Koordinatensystems
x_g	–	X-Achse des geodätischen Koordinatensystems
y_E	–	Y-Achse des erdfesten Koordinatensystems
y_a	–	Y-Achse des aerodynamischen Koordinatensystems
y_f	–	Y-Achse des flugzeugfesten Koordinatensystems
y_g	–	Y-Achse des geodätischen Koordinatensystems
z_E	–	Z-Achse des erdfesten Koordinatensystems
z_a	–	Z-Achse des aerodynamischen Koordinatensystems
z_f	–	Z-Achse des flugzeugfesten Koordinatensystems
z_g	–	Z-Achse des geodätischen Koordinatensystems

Griechische Symbole

Symbol	Einheit	Bedeutung
α	$^{\circ}$, rad	Anstellwinkel
α_{Htail}	$^{\circ}$, rad	Anstellwinkel Höhenleitwerk
α_{Wing}	$^{\circ}$, rad	Anstellwinkel Flügel
α_S	$^{\circ}$, rad	Höhenwinkel der Sonne
β	$^{\circ}$, rad	Schiebewinkel
γ	$^{\circ}$, rad	Bahnneigungswinkel
$\dot{\gamma}$	rad/s	Nickrate
δ	$^{\circ}$, rad	Deklination der Sonne
ε_{dir}	W/m^2	direkte Sonnenstrahlung
ε_{min}	—	Gleitverhältnis für bestes Gleiten
ε_t	$^{\circ}$, rad	Verwindung des Flügels
ε_0	W/m^2	mittlere Solarkonstante
$\varepsilon_0(n)$	W/m^2	mittlere Solarkonstante abhängig vom Jahrestag
η_A	—	Antriebseffizienz
$\eta_{el Mot}$	—	elektr. Wirkungsgrad
η_{Prop}	—	Wirkungsgrad des Propeller
η_{SG}	—	Gesamtwirkungsgrad des Solargenerators
$\eta_{SP i}$	—	Wirkungsgrad des Solarpanels i
ζ_{dgr}	—	Degradierungsfaktor
$\zeta_{dgr n}$	—	Degradierungsfaktor nach n Zyklen
$\zeta_{dgr 0}$	—	Degradierungsfaktor bei Missionsstart
θ	$^{\circ}$, rad	Nickwinkel
θ_{SP}	$^{\circ}$, rad	Winkel der Panelorientierung um die Y-Achse
Λ	—	Streckung
λ	$^{\circ}$, rad	Längengrad
$\nu/\nu(h)$	m^2/s	kinematische Viskosität
ξ	$^{\circ}$, rad	Zenitwinkel der Sonne/ Sonnenhöhe
ξ_{dep}	$^{\circ}$, rad	Depression der Sonne
ρ_0	kg/m^3	Luftdichte auf Meereshöhe
$\rho(h)$	kg/m^3	Luftdichte
σ_n	—	Degradierungsrate
τ	$^{\circ}$, rad	Stundenwinkel
$\varphi_{l/4}$	$^{\circ}$, rad	Pfeilung des Flügels
ϕ	$^{\circ}$, rad	Breitengrad
ϕ_{roll}	$^{\circ}$, rad	Rollwinkel
ϕ_{SP}	$^{\circ}$, rad	Winkel der Panelorientierung um die X-Achse
Ψ	$^{\circ}$, rad	Gierwinkel
Ψ_S	$^{\circ}$, rad	Azimut der Sonne
Ω_E	$^{\circ}/s$, rad/s	Umlaufgeschwindigkeit der Erde

Abkürzungen

Abkürzung	Bedeutung
APD	Aircraft Preliminary Design
BAE	British Aerospace Electronic Systems
CDR	Critical Design Review
CoG	Center of Gravity
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EAS	Equivalent Airspeed
GaAs	Gallium Arsenide
GUI	Graphical User Interface
HALE	High Altitude Long Endurance
HAP	High Altitude Platform
HAPS	High Altitude Pseudo Satellite
HIL	Hardware-in-the-Loop
HLW	Höhenleitwerk
ICAO	International Civil Aviation Organization
KOS	Koordinatensystem
Lead-acid	Bleiakkumulator
Li-Ion	Lithium Ionen
Li-O ₂	Lithiumoxid
Li-S	Lithium-Schwefel
Li-Si	Lithium-Silizium
Ltd.	Limited
MSL	Mean Sea Level
Ni-CD	Nickel-Cadmium
Ni-MH	Nickel-Metallhydrid-Akkumulator
OOP	objekt-orientierte Programmierung
PDR	Preliminary Design Review
PHASA	Persistent High Altitude Solar Aircraft
RADAR	Radio Detection and Ranging
SAR	Synthetic Aperture Radar
SL	Sea Level
SLW	Seitenleitwerk
SoC	State of Charge
SSR	System Specification Review
TAS	True Airspeed
TRR	Test Readiness Review
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language

1 Einführung

1.1 Motivation

Zu Beginn einer Flugzeugentwicklung werden Ziele festgelegt, die ein Fluggerät während seiner Mission erreichen soll. Folglich resultieren daraus Kenngrößen für das Flugzeug, die nach verschiedenen Methoden geschätzt oder berechnet werden. Im Projekt High Altitude Platform, im Folgenden HAP, des Deutschen Zentrums für Luft- und Raumfahrt (DLR) wurden Ziele für ein solarelektrisches Stratosphärenflugzeug festgelegt. Mehrere Institute des Forschungszentrums arbeiten an verschiedenen Standorten gemeinsam an der Ausarbeitung des Flugzeugentwurfes und der einzelnen Komponenten. Abbildung 1.1 zeigt ein erstes Entwurfsmodell des geplanten Flugzeuges.



Abbildung 1.1: Erstes Entwurfsmodell HAP [1]

Im Zuge der Auslegung einer Komponente ergeben sich seine Kenngrößen wie beispielsweise Masse, Abmessung, Energiedichte und Wirkungsgrad. Generell besteht eine Differenz zu zuvor projektierten Parametern. Summierend auf das Flugzeug entstehen signifikante Unterschiede, welche die übergeordneten Ziele der Flugleistungen negativ beeinflussen können. Zur Absicherung dieser kann eine Simulation des Flugzeuges innerhalb seiner geplanten Umgebung und durchzuführenden Mission erfolgen. Dies erlaubt im Voraus Einfluss auf kritische Komponenten zu nehmen und folglich ungeplante Kosten zu reduzieren bzw. sogar zu vermeiden. Weiterhin wird transparent dargestellt, ob das Flugzeug die geplante Mission erfolgreich absolvieren kann.

Der Bedarf dieser Arbeit besteht im Hinblick auf das anstehende Critical Design Review (CDR) Anfang 2022 im Projekt HAP. Systemkomponenten des Flugzeuges erhalten erste greifbare Daten, die eine Verifikation der Flugleistungen erfordern.

1.2 Ziel der Arbeit

Ziel der Ausarbeitung ist die Erstellung einer Missionssimulation zur Bewertung der Flugzeugkonfiguration HAP. Die Umsetzung soll mit der Software Matlab in objekt-orientierter Programmierung (OOP) erfolgen. Die Objektorientierung bietet den Vorteil, dass Komponenten des Flugzeuges als Objekt angelegt werden können und diese objektspezifische Funktionen erhalten. Dies resultiert in einem vereinfachten Handling und erleichtert die Erweiterung des Programms.

Neben dem Flugzeug und seiner Komponenten ist auch die geplante Mission und seine Umgebung abzubilden. Letzt genanntes enthält wiederum atmosphärische Größen, Winddaten und die Leistungsdaten der Sonne. Die Ergebnisse der Simulation sollen mehrere Zwecke erfüllen. Primär sind Aussagen zur Stationierbarkeit bereitzustellen. Zusätzlich ist es notwendig kritische Komponenten des Flugzeuges aufzuzeigen. Parameteranalysen können als Ergänzung erarbeitet werden.

1.3 Gliederung der Arbeit

Der Aufbau der weiterführenden Kapitel lehnt sich an die Erstellung des Simulationsprogrammes und dessen Nutzung an. Da die Simulation in Englisch aufgebaut wird, entsprechen sämtliche Klassen- und Funktionsnamen sowie Indizes englischen Begriffen. Im Folgenden wird jedes Kapitel inhaltlich erläutert:

In **Kapitel 1 “Einführung”** wird die Motivation des Themas, die Ziele und der Aufbau der vorliegenden Arbeit erklärt.

Kapitel 2 “Stand der Technik” beschreibt am Markt vorhandene hochfliegende Solarplattformen sowie Schlüsselkomponenten und vorhandene Software zu flugzeugbezogenen Simulatoren.

Kapitel 3 “Struktur und Aufbau der Simulation” stellt einen Überblick zum Aufbau der Simulation vor. Es wird die Klassen- und Objektstruktur beschrieben inklusive aller Klassen und ihrer jeweiligen Funktionen.

In **Kapitel 4 “Modellierung des Flugzeuges HAP und seiner Umgebung”** wird die Berechnungsgrundlage der Klassen dokumentiert. Des Weiteren werden verwendete Formeln aus entsprechender Literatur dargelegt und/oder hergeleitet.

In **Kapitel 5 “Verifikation der Simulation”** wird beschrieben, wie die Simulation bzw. Teile davon verifiziert werden können. Des Weiteren wird die Verifizierung weiterer Flugzeugtypen erläutert.

Kapitel 6 “Simulationsergebnisse zum Projekt HAP” legt die Missionsanalyse zum aktuellen Stand des geplanten Flugzeuges HAP dar. Ausgehend von einer definierten Standardmission werden Analysen beschrieben und Optimierungen der Mission vorgestellt. Darüber hinaus wird eine Erweiterung des Programms zur Durchführung von Studien vorgestellt.

Abschließend wird in **Kapitel 7 “Zusammenfassung und Ausblick”** die Ausarbeitung rückwirkend betrachtet und es werden mögliche Ansätze zur Erweiterung der Themenstellung dargelegt.

Inhalte der Anhänge

In **Anhang A** finden sich die für die Simulation verwendeten Parameter der Flugzeuge, Missionsdaten und Formeln. **Anhang B** beschreibt weitere Solarflugzeuge als zusätzliche Ergänzung zum Stand der Technik. **Anhang C** listet rein informativ den aktuellen Verlauf im Projekt HAP auf. **Anhang D** beinhaltet eine Parameterliste zum Programm, weitere Analysen sowie Programmblöcke, die im Zuge der Entwicklung des Simulators entfallen sind bzw. ersetzt wurden.

2 Stand der Technik

2.1 Hochfliegende Solarplattformen

Die Simulation wird für elektrisch angetriebene Solarflugzeuge ausgelegt, die als sogenannte HALE-Fluggeräte bezeichnet werden. HALE steht für High Altitude Long Endurance und charakterisiert Flughöhen im Bereich von 15 bis 20 km, sowie Missionszeiten über 24 Stunden [2]. Im Folgenden werden neben dem Flugzeug HAP konfigurations- und leistungsähnliche HALE-Flugzeuge vorgestellt. Weitere Flugzeuge anderer Konfiguration und/oder Leistung sind in Anhang B beschrieben.

2.1.1 HAP

Das Flugzeug HAP ist Teil eines Querschnittprojektes innerhalb des DLR und mit klassischer Flügel-Leitwerkskombination entworfen. Diese Kombination wird in der Fachsprache auch Drachenkonfiguration genannt. Die Hauptaufgabe des Flugzeuges dient der Platzierung und Positionierung verschiedener Nutzlasten im Bereich der Stratosphäre. Weiterhin sollen Kompetenzen und Erfahrungen in den Fachbereichen des DLR aufgebaut werden. Die geplante Masse beträgt 138 kg bei einer Spannweite von 27 m. Die Reise Flughöhe liegt bei ca. 20 km bei einer Fluggeschwindigkeit von ca. 10 m/s Equivalent Airspeed (EAS). Als Batteriesystem werden Lithium-Ionen (Li-Ion) Batterien vorgesehen, welche mit Gallium-Arsenid (GaAs) Solarzellen gekoppelt sind. Ein erster Technologiedemonstrator und der Erstflug ist für Ende 2022 geplant. Zwei verschiedene Nutzlasten mit je maximal 5 kg sind geplant. Diese bestehen entweder aus einem Kamera- oder SAR-System und dienen beide der Beobachtung und Abtastung größerer Gebiete [1]. Ein SAR-System ist ein abbildendes RADAR-System. Das geplante Flugzeug wurde bereits auf Abbildung 1.1 dargestellt.

2.1.2 BAE PHASA 35



Abbildung 2.1: Digitales Entwurfsmodell PHASA-35 [5]

Ein weiteres solarbetriebenes Flugzeug ist PHASA 35, entstanden durch eine Kooperation der Firmen BAE Systems und Prismatic Ltd.. PHASA ist die Abkürzung für Persistent High Altitude Solar Aircraft und wirbt durch die Bezeichnung für Nachhaltigkeit. Getrieben durch großes Interesse seitens

des britischen und amerikanischen Militärs konnte der Erstflug im Jahr 2020 stattfinden. In Simulationen konnte das Flugzeug 72 Stunden im Einsatz in der Stratosphäre verbleiben. Die Abflugmasse beträgt 150 kg bei 35 m Spannweite. Die Reisegeschwindigkeit liegt im Bereich von 93 bis 145 km/h in einer Flughöhe von 15 bis 21 km. Es wird mit Nutzlasten bis 15 kg geworben für Anwendungen der Überwachung oder dem Mobilfunk [3] [4] [6]. Abbildung 2.1 zeigt ein Modell des Flugzeuges.

2.1.3 Zephyr S

Der Flugzeugbauer Airbus stellte mit dem Zephyr S im Jahr 2018 bereits beim Erstflug einen Rekord über knapp 26 Tage Dauerflug auf. Das Flugzeug besitzt eine Masse von 75 kg bei einer Spannweite von 25 m. Die Reishöhe liegt bei 21,3 km in der eine Nutzlast von 5 kg transportiert wird. Eine Besonderheit ist die Verwendung von Lithium-Silizium Batterien, die hohe Energiedichten vermuten lassen. Außerdem ist das Höhen- und Seitenleitwerk zusätzlich zum Flügel mit Solarzellen bestückt. Neben der Anwendung als mobile Internetplattform wirbt Airbus mit weiteren optischen und RADAR-gestützten Optionen. [8] [10] [11]. Abbildung 2.2 zeigt den Zephyr S im Abflug während der Flugkampagne 2021.



Abbildung 2.2: Das Solarflugzeug Zephyr S beim Start [9]

2.1.4 Anforderungen Solarflugzeuge

Die zuvor vorgestellten Flugzeuge können mit ihren Entwurfsparametern in ein entsprechendes Diagramm eingeordnet werden, um sie besser bewerten zu können. Abbildung 2.3 zeigt, dass zwischen dem Flugzeug HAP und der BAE PHASA 35 in Bezug auf den Leistungsparameter nur eine Differenz von ca. 0,4 W/N vorliegt. Somit sollten diese annähernd gleiche Flugleistungen im Steigflug erreichen können. Der Zephyr S ist besonders auffällig durch die wesentliche geringere Flächenbelastung welche aus konsequenter angewendetem Leichtbau resultiert. Parallel dazu könnte weniger Systemmasse wie beispielsweise Batteriemasse verbaut sein. Des Weiteren sollte das erhöhte Schub-Gewichtsverhältnis zu besseren Steigeigenschaften führen. Die Berechnungen der Entwurfsparametern aus Abbildung 2.3 sind in Anhang A.4.3 zu finden.

Im Vergleich mit bemannten Flugzeugen ähnlicher Dimension wird der Unterschied bzgl. des Gewichts durch die höhere Flächenbelastung deutlich. In Abbildung 2.4 ist dies durch weitere Flugzeuge dargestellt (Daten- und Bildquellen aus [13] [14] [15] [16], Berechnungen siehe Anhang A.4.3).

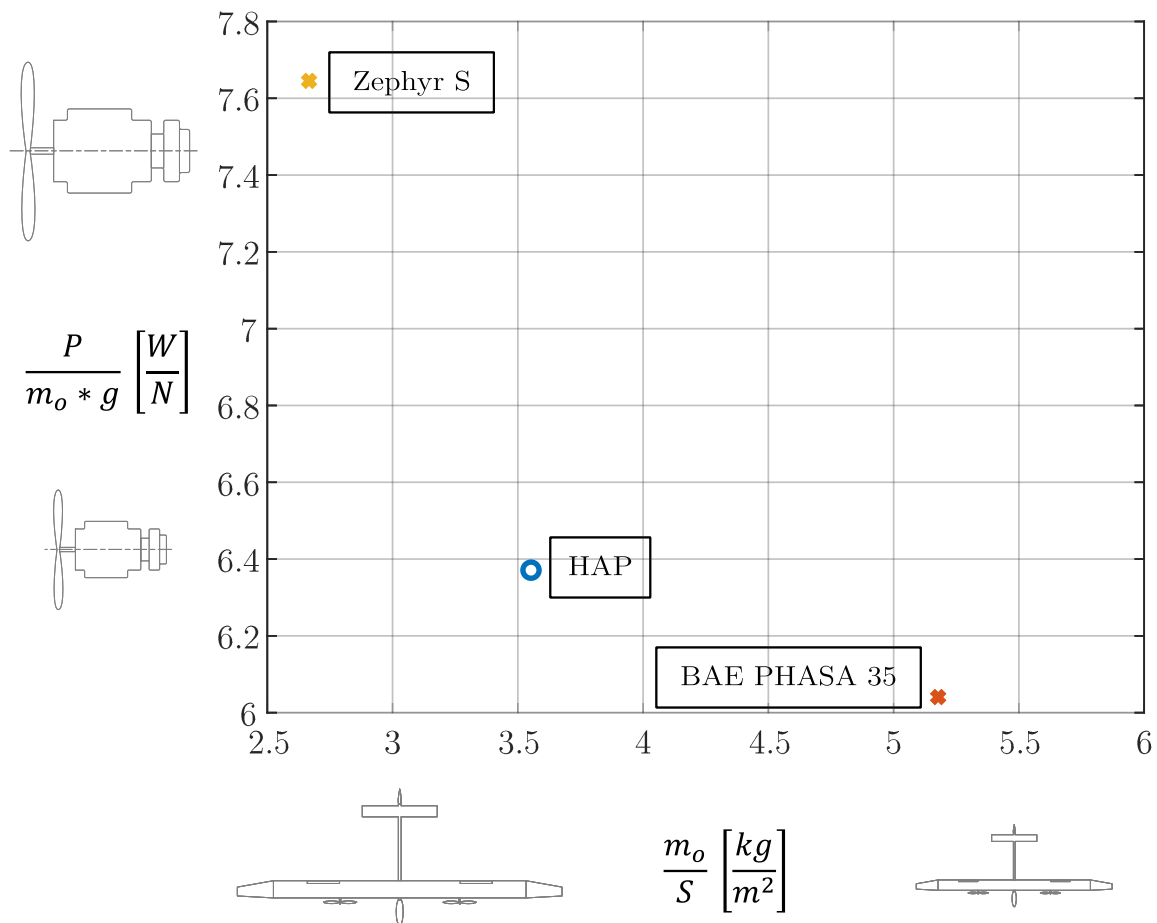


Abbildung 2.3: Entwurfspunkte für HALE Flugzeuge

Das in der Abbildung gezeigte Solarflugzeug Icare II konnte bei einer Flächenbelastung von $13,76 \text{ kg/m}^2$ ca. 350 km weit fliegen, was bei einer Geschwindigkeit von 60 km/h auf ca. $5\text{-}6 \text{ h}$ Flugzeit schließen lässt [16]. Danach waren die eingebauten Akkumulatoren entladen bzw. die Solarleistung nicht mehr ausreichend. Die Flugzeuge eGenius und ASMe33 gehören in die Kategorie angetriebener Segelflugzeuge. Die wesentlich höhere Flächenbelastung erlaubt, abhängig vom Wind, nur einige Stunden Flugbetrieb ohne Zwischenlandung. Es ist anzumerken, dass diese Flugzeuge keine Möglichkeit zur Nutzung solarer Energie besitzen.

Der Vergleich dieser elektrisch angetriebenen Flugzeuge verdeutlicht die Abhängigkeit einer geringen Flächenbelastung von der maximal möglichen Flugdauer. Daher ist die Einhaltung einer geringen Flächenbelastung zwingend erforderlich zur Erfüllung des geforderten Missionsprofils über mehrere Wochen.

2.2 Technologiestand

Neben der erwähnten Flächenbelastung sind noch weitere Faktoren entscheidend für eine dauerhafte Platzierung des Fluggeräts in der Stratosphäre. Dazu zählen unter anderem die Solarzellen und die Batterien.

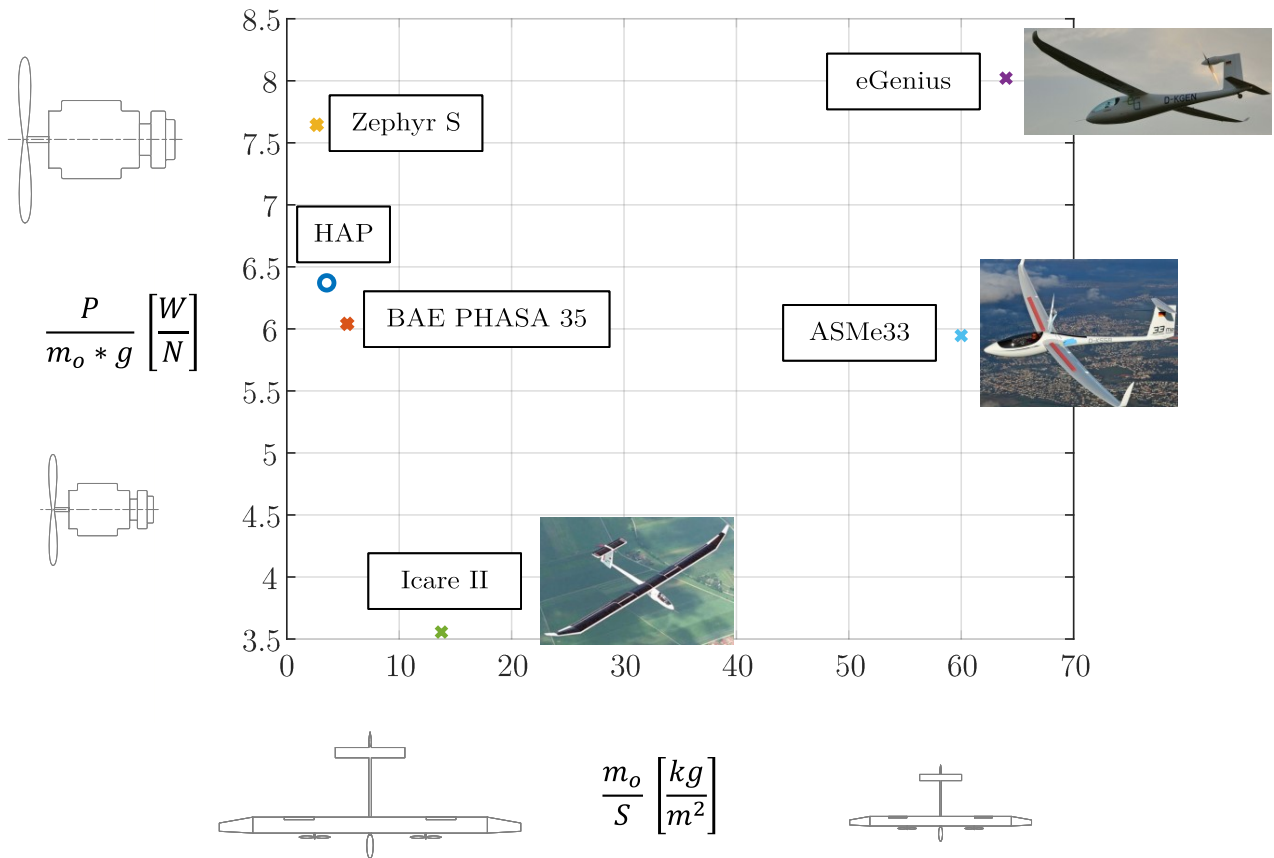


Abbildung 2.4: Entwurfspunkte für verschiedene Flugzeuge

Solarzellen werden durch zwei Kennwerte charakterisiert. Zum einen den Wirkungsgrad, der angibt wieviel der solaren Strahlung in nutzbare Energie umgewandelt werden kann. Zum anderen betrifft es die spezifische Masse, welche die Masse bezogen auf die Solarfläche ausdrückt. In Tabelle 2.1 sind für die Anwendung von HALE Flugzeugen in Betracht zu ziehende Solarzellen inklusive ihrer Kennwerte aufgeführt. Die Wahl der Solarzelle wird meist abhängig von ihrem Preis getroffen. Beispielsweise kann der Preis für eine Tripelzelle GaAs bereits mehrere zehntausend Euro pro Quadratmeter betragen [1].

Art	Wirkungsgrad	Spez. Masse
	[%]	[g/m ²]
Monolithisch GaAs	25	350
Tripelzelle GaAs	28	330
Kupfer Indium GaAs	12	350
Monokristallines Silizium	21	650

Tabelle 2.1: Verschiedene Solarzellen mit Kenndaten [17]

Der aktuelle Trend der Batterietechnologien aufgrund des politischen Wandels zur Elektromobilität ist vielversprechend [18]. Batterien werden nach ihrer Energiedichte in Wh/kg charakterisiert. Steigende Energiedichten ermöglichen mehr Kapazitäten bei geringerem Gewicht. Folglich sinkt die Flächenbelastung des Flugzeuges. In Abbildung 2.5 sind verschiedene Batterietypen nach ihrer volumetrischen

und gravimetrischen Energiedichte unterteilt dargestellt. Die darin gezeigte Li-Ion Technologie ist mittlerweile vielfach verwendet, jedoch zielt der Trend zu höheren Energiedichten. Für die Wahl der Batteriezellen ist oftmals noch der Preis pro Kilogramm Batterie entscheidend. Allerdings könnte die anhaltende Entwicklung zur Elektromobilität Preise drastisch reduzieren.

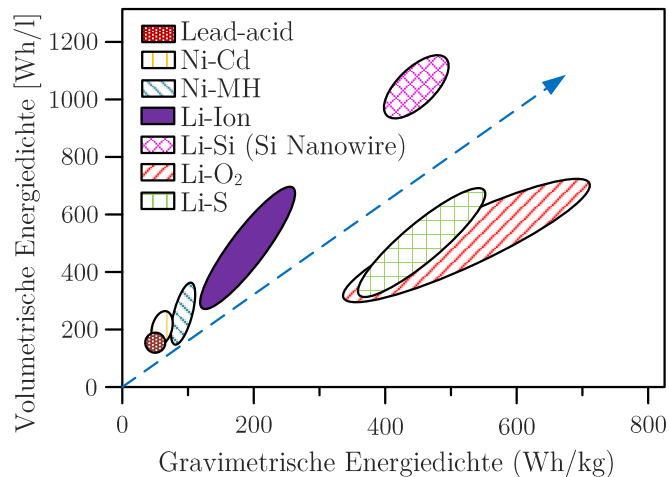


Abbildung 2.5: Aktuelle Batteriedichten 2021 [17]

2.3 Bedarf der Simulation

Auf Basis der projizierten Technologien für das Flugzeug HAP kann berechnet werden wo und in welchen Zeiträumen eine dauerhafte Stationierung des Demonstrators möglich ist. In Abbildung 2.6 ist dies wiedergegeben. Dabei werden die Solarzellen, sowie aerodynamische Kräfte und die Kennlinien der verwendeten Propeller berücksichtigt. Es ist deutlich erkennbar, dass eine Stationierung in den oberen Breitengraden der Erde zu Jahresbeginn und -ende nur durch sehr geringe Flächenbelastung möglich ist.

Für das Projekt HAP wurde eine Flächenbelastung von $3,7 \text{ kg/m}^2$ projiziert. Unter Berücksichtigung der Energiedichten der Batterien ergibt sich Abbildung 2.7. Daraus resultiert eine zusätzliche Einschränkung der Einsatzgebiete und -zeiträumen abhängig von der gewählten Batterietechnologie.

Aus diesen Analysen ergibt sich, dass Abweichungen des Flugzeuges vom zuvor projizierten Stand signifikante Auswirkungen auf die dauerhafte Stationierung bewirken können. Mitunter verstärkt sich dieser Effekt durch Abhängigkeiten einiger Parameter untereinander, wodurch einfache Voraussetzungen sehr umfangreich werden.

Gegenüber den bisher rein technisch beschriebenen Aspekten ist eine betriebswirtschaftliche Betrachtung sinnvoll. Eine Änderung von Komponenten während eines Projektes kann je nach Reifegrad des Designs zu unerwartet hohen Kosten führen. Abbildung 2.8 zeigt darüber hinaus, dass die Möglichkeiten zur Einflussnahme zum Projektende hin exponentiell sinken.

Aus den genannten Gründen ist ein Missionssimulator sinnvoll. Mit Hilfe dessen können Analysen schnell und einfach durchgeführt werden. Dies gilt insbesondere zur Feststellung des Missionserfolges aufgrund abweichender Parameter. Darüber hinaus können kritische Komponenten schneller erkannt

werden und gegebenenfalls vorzeitig überarbeitet oder ersetzt werden.

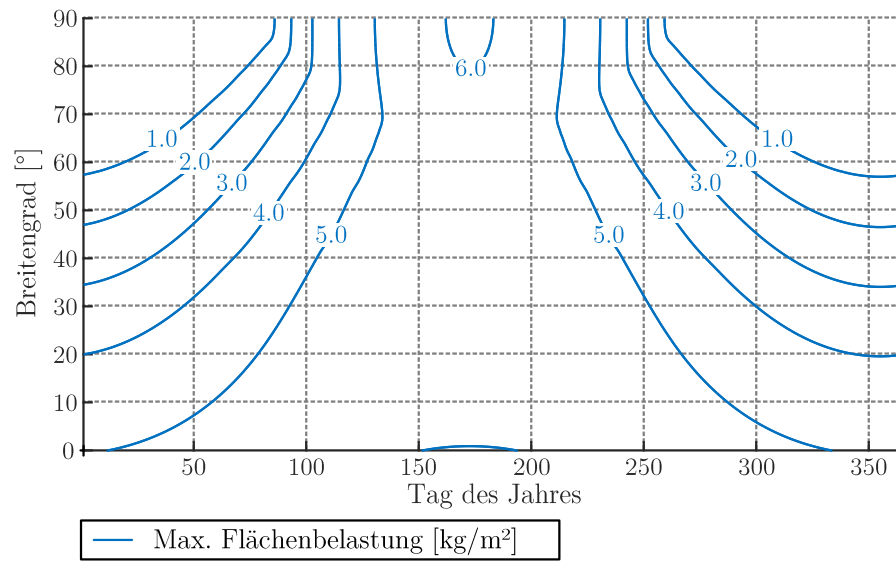


Abbildung 2.6: Dauerhafte Stationierung abhängig von der Flächenbelastung [17]

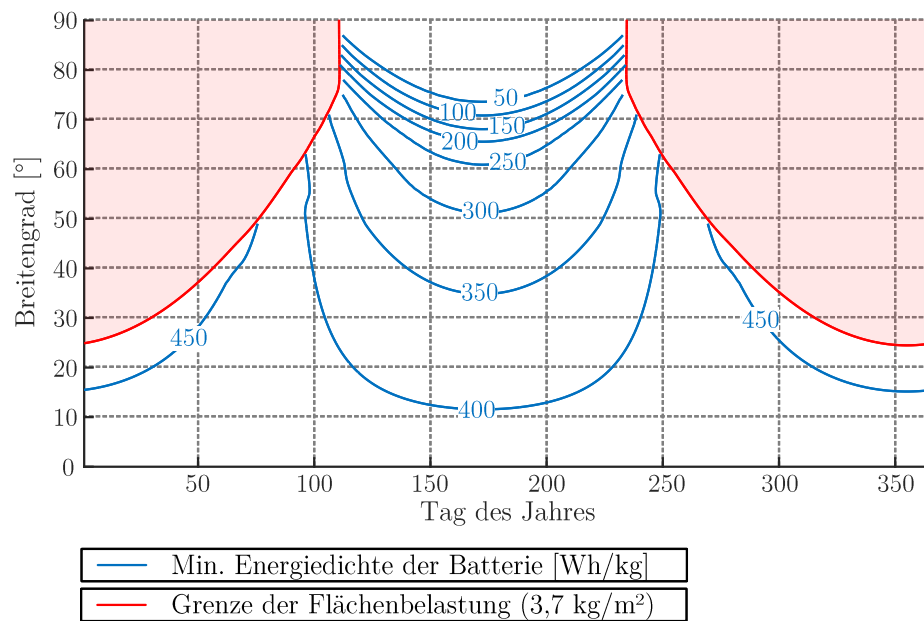


Abbildung 2.7: Dauerhafte Stationierung abhängig von der Energiedichte der Batterie [17]

2.4 Vorhandene Simulationssysteme

Die Analyse von Flugleistungen und Missionsprofilen wird bereits teilweise durch am Markt vorhandene Systeme realisiert. Dabei sind diese Funktionen meist als Erweiterung von Vorentwurfssystemen integriert. Nachfolgend werden Softwareanwendungen beschrieben die erweiterte Funktionalitäten zur Durchführung von Missionsanalysen aufweisen.

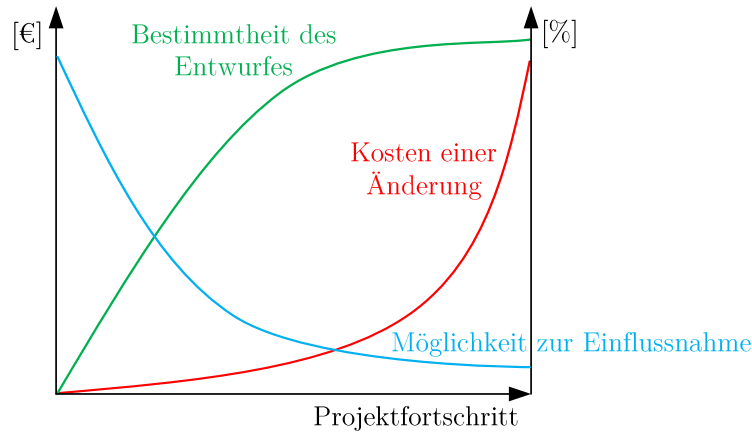


Abbildung 2.8: Kosten einer Änderung im Projektfortschritt nach [19]

2.4.1 Pacelab APD

Das Flugzeugvorentwurfstool Pacelab APD der Firma PACE, Berlin, bietet eine Plattform mit vielen Konfigurations- und Analysemöglichkeiten. Die Abkürzung APD steht für Aircraft Preliminary Design [20]. Das Produkt wirbt speziell für den Einsatz in der frühen Entwurfsphase. Eine Besonderheit ist die Bereitstellung offener Schnittstellen, sodass der Funktionsumfang eigenständig mit individuellen Methoden, Simulations- und/oder Analysewerkzeugen erweitert werden kann [21]. Dadurch könnte beispielsweise auf die vorhandenen Entwurfsmethoden zurückgegriffen werden um diese durch eine eigene Missionssimulation zu ergänzen.

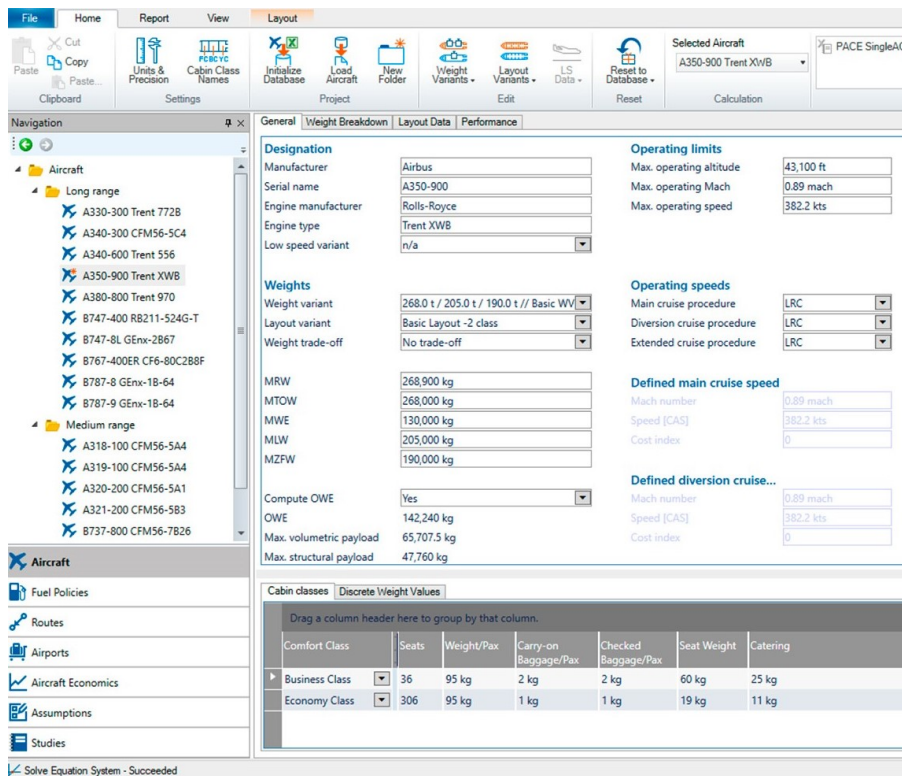


Abbildung 2.9: Eingabemaske Pacelab Mission Suite [22]

Hinsichtlich Analyse von Missionen bietet das Modul Pacelab Mission Suite die passendste Affinität. Das Programm spezialisiert sich auf Flugstreckenanalysen von Verkehrsflugzeugen. Dabei wird besonders die Optimierung des Kraftstoffverbrauchs beworben unter Berücksichtigung von aktuellen Wetterdaten [22]. In Abbildung 2.9 ist der Ausschnitt des Graphical User Interface (GUI) des Programms gezeigt. Die Parameter sind sehr übersichtlich sortiert und werden durch Dropdown Menüs unterstützt.

2.4.2 Aerospace Blockset

Das Produkt Aerospace Blockset der Firma Mathworks, Vereinigten Staaten von Amerika, bietet Referenzbeispiele und -blöcke zur Modellierung und Simulation von Luft- und Raumfahrzeugen. Die Programmbeispiele werden mit dem Matlab Modul Simulink als Blockschaltbilder ausgeführt. Es beinhaltet validierte Modelle der Flugumgebung sowie Bausteine für Antrieb und Aktuator-Dynamiken. Weiterhin sind Pilotenmodelle und Triebwerksysteme hinterlegt. Durch Koordinatentransformationen kann auch die Fluglage und -orientierung des Luftfahrzeuges simuliert werden. Ergebnisse können mehrdimensional ausgegeben werden. [23]

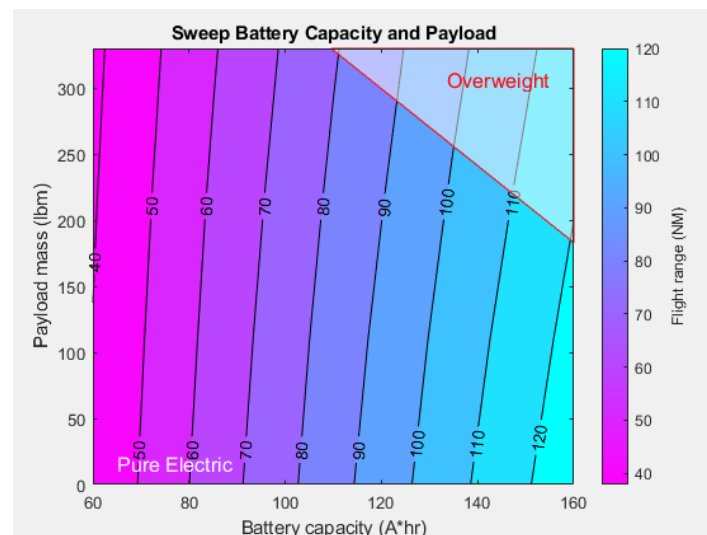


Abbildung 2.10: Reichweitenanalyse eines Hybridflugzeuges [25]

Neben Flug- und Missionsanalysen sind auch Konzeptstudien möglich. Weiterhin sind Software Integrations- und Hardware-in-the-Loop-Tests (HIL) umsetzbar. Letztere sind interessant für die Simulation erster Komponenten. Darüber hinaus sind anwendungsspezifische Programmierbeispiele zur Einarbeitung in die Software gegeben. Zur Thematik passende Beispiele sind das Umgebungsmodell und die Flugvisualisierung [24]. Abbildung 2.10 zeigt das Ergebnis einer Parameteranalyse, die mithilfe des Moduls erstellt wurde. Darin wird die Abhängigkeit der Flugreichweite von der Batterie- und Nutzlastmasse visualisiert.

Das Modul bietet großes Potential für detaillierte Missionsanalysen. Nachteilig ist jedoch, dass lediglich Beispielblöcke gegeben werden und der Umfang zur Einarbeitung hoch ist.

3 Struktur und Aufbau der Simulation

3.1 Grundlagen objektorientierte Programmierung

Um die Beschreibung des Simulationsprogramms zu erleichtern, ist es notwendig den Begriff der Objektorientierung und die damit verbundene Programmstruktur zu definieren und aufzuschlüsseln. In der Programmierung werden verschiedenste Programmierstile, auch Programmierparadigmen genannt, unterschieden. Im Anwendungsfall einer Missionssimulation kann primär zwischen der sequentiellen und der objektorientierten Programmierung unterschieden werden [26]. Erst genannte arbeitet ein Programm hierarchisch meist von oben nach unten ab und erfüllt zeilenweise die darin enthaltenen Aufgaben. Dabei sinkt die Übersichtlichkeit und folglich Bearbeitbarkeit mit steigender Anzahl der Programmzeilen stetig an, was wiederum zu weiteren Nachteilen führt. Ein zu nennender Nachteil ist die Zeit, die ein neuer Programmierer benötigt, um sich in das Programm einzuarbeiten um eventuelle Änderungen und/oder Erweiterungen vorzunehmen [27].

Bei der objektorientierten Programmierung wird die Programmarchitektur entsprechend der Komponenten und Bauteile der jeweiligen Anwendung aufgebaut. Die Umsetzung erfolgt mithilfe der sogenannten Objekte. Beispiele für Objekte sind der Flugzeugflügel, die Atmosphäre oder die geplante Mission. Die Interaktion der Objekte untereinander bildet die eigentliche Ausführung des Programmes ab. Dies ist möglich durch die Attribute und Funktionen der Objekte. Die Attribute eines Objektes beschreiben üblicherweise objektspezifische Eigenschaften. Neben physikalischen Werten können hier auch Namen, Kennfelder oder sogar weitere Objekte zugeordnet werden. Die Funktionen bilden die Steuerelemente eines Objektes. Sie werden auch als Methoden bezeichnet. Durch diese kann ein Objekt Eigenschaften ändern, mit anderen Objekten interagieren und programmtypische Aufgaben ausführen [28]. Ein weiterer Vorteil besteht bei der Übergabe von Daten an andere Objekte. Meist werden hier komplette Objekte selbst übergeben. Wird ein Objekt im Nachhinein um ein Attribut oder eine Funktion erweitert, ist meist keine Änderung in anderen Programmzeilen notwendig. Ist der Aufbau der Objekte intuitiv und praxisnah umgesetzt, können Änderungen und Erweiterungen am Programm sehr einfach vorgenommen werden.

Es ist noch anzumerken, dass Objekte jeweils aus einer Klasse gebildet werden. Das heißt eine Klasse liefert das leere Konstrukt, dass mit den spezifischen Daten des jeweiligen Objektes gefüllt wird. So lassen sich mehrere Objekte mit unterschiedlichen Eigenschaften jedoch gleichen Funktionen erstellen, beispielsweise der Flugzeugflügel und das Höhenleitwerk. Die Erstellung eines Objektes wird als Instanziierung bezeichnet [28].

Die Darstellung einer Klasse und/oder eines Objektes inklusive seiner Attribute und seiner Funktionen erfolgt in Dokumentationen über UML Klassendiagramme. UML steht hier für Unified Modeling Language [29]. Abbildung 3.1 dient als einfaches Beispiel zur Erläuterung einer Klasse. Im oberen Teil steht der Klassenname. In diesem Beispiel umfasst die Klasse alle flügelähnlichen Komponenten. Der mittlere Block enthält alle Attribute und der untere die Funktionen. Eine genauere Beschreibung der Attribute sollte mindestens durch Kommentare innerhalb des Programmes erfolgen. Durch Linien und Pfeile zu anderen Objekte werden Verbindungen und Hierarchien aufgezeigt. Diese vereinfachte Darstellung der Objekte ist besonders wichtig um die Übersicht während des Programmierens zu wahren. Die stetige Pflege der Klassendiagramme ist zu empfehlen.

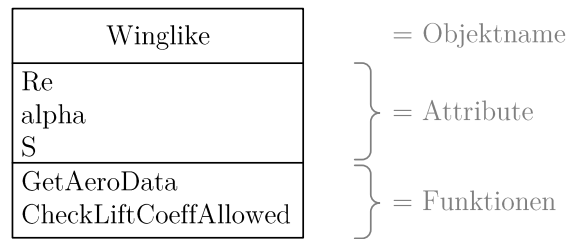


Abbildung 3.1: Beispieldarstellung einer Klasse

In den nachfolgenden Bezeichnungen der Klassen, Attribute und Funktionen werden englische Begriffe und Abkürzungen verwendet, da die gesamte Simulation englischsprachig aufgebaut wurde. Diagramme und Bezüge sind lediglich den in der Informatik üblichen Nomenklaturen angelehnt. Formelle Abweichungen sind bewusst gewählt um einen praktischen Bezug zum Programm herzustellen. Funktionen eines Objektes werden immer wie folgt angegeben. Zuerst wird der Objektname geschrieben gefolgt durch den Namen der Funktion. Ein Beispiel hierzu ist **Wing.SetData**. Innerhalb des Objektes Wing wird die Funktion SetData aufgerufen.

3.2 Aufbau der Programmstruktur

Die Struktur der Simulation basiert auf einem zentralen Objekt in dem die Hauptschritte zur Simulation abgehandelt werden. Das Objekt und die dazugehörige Klasse wird als **Scenario** bezeichnet. Es erzeugt während seiner Instanziierung weitere Objekte aus den Klassen **Aircraft**, **Mission**, **Environment** und **SimLog**. Die Klasse Aircraft handelt sämtliche Aktionen betreffend des Flugzeuges ab. Weiterhin erzeugt sie ihr untergeordnet weitere Unterklassen, die die Hauptkomponenten des Flugzeuges abbilden. Diese bestehen aus den Klassen **Winglike**, **DragProducer**, **Battery**, **SolarGenerator**, **Propulsion**. Der Klasse Propulsion sind die Klassen **Propeller** und **Motor** untergeordnet. Die Klasse Mission umfasst alle Daten und Funktionen bezüglich der Bahnplanung. Darunter fallen geometrische Positionsdaten als auch zeitbezogene Daten. Die Klasse Environment beschreibt die Umgebung um das Flugzeug. Sie generiert die Klassen **Atmosphere**, **Sun** und **Wind** für entsprechende Funktionen. Die letzte Klasse SimLog dient der Datenerfassung und -ausgabe. Sie wird zwar dem Scenario untergeordnet erzeugt, ist aber von ihrer Wertigkeit gleich wie diese anzusetzen.

Eine Übersicht der gesamten Klassenstruktur wird in Abbildung 3.2 gegeben. Die darin gezeigten Level dienen der besseren Visualisierung und Dokumentation der Struktur. Die Klasse Scenario und SimLog bilden die Top Level Klassen. Aircraft, Mission und Environment umfassen den darunter geordneten Mid Level. Die restlichen Klassen gehören in den Sub Level. Dem Level aufsteigend steigt die jeweilige Komplexität der Funktionen der dazugehörigen Klasse.

Die Simulation wird durch das Startskript **PerformSimulation** ausgelöst. Darin wird die Klasse Scenario als gleichnamiges Objekt instanziiert. Bei der Erstellung des Scenario werden die Objekte aus den Mid Level Klassen ebenfalls gleichnamig erzeugt, sowie das Objekt SimLog der SimLog Klasse. Alle Objekte des Sub Level bis auf die des Typs Winglike und DragProducer werden gleichermaßen instanziiert. Mit der Klasse Winglike wird das Objekt **Wing** und **HTail** erzeugt entsprechend dem Hauptflügel und dem Höhenleitwerk. Dabei wird erstmalig ein bedeutender Vorteil der Objektorientierung deutlich. Beide Objekte besitzen unterschiedliche Attribute, jedoch können sie auf gleiche Funktionen zugreifen. Dies reduziert zusätzlich den Programmieraufwand. Aus der Klasse DragProducer entstehen die Objekte **VTail**, **Fuselage**, **LandingGear** und **Antennas**. Diese sind jeweils gleichbedeutend mit dem Seitenleitwerk, Rumpf, Fahrwerk und Antennenanbauten. Die komplette Instanziierung ist in Abbildung 3.3 nochmals gezeigt.

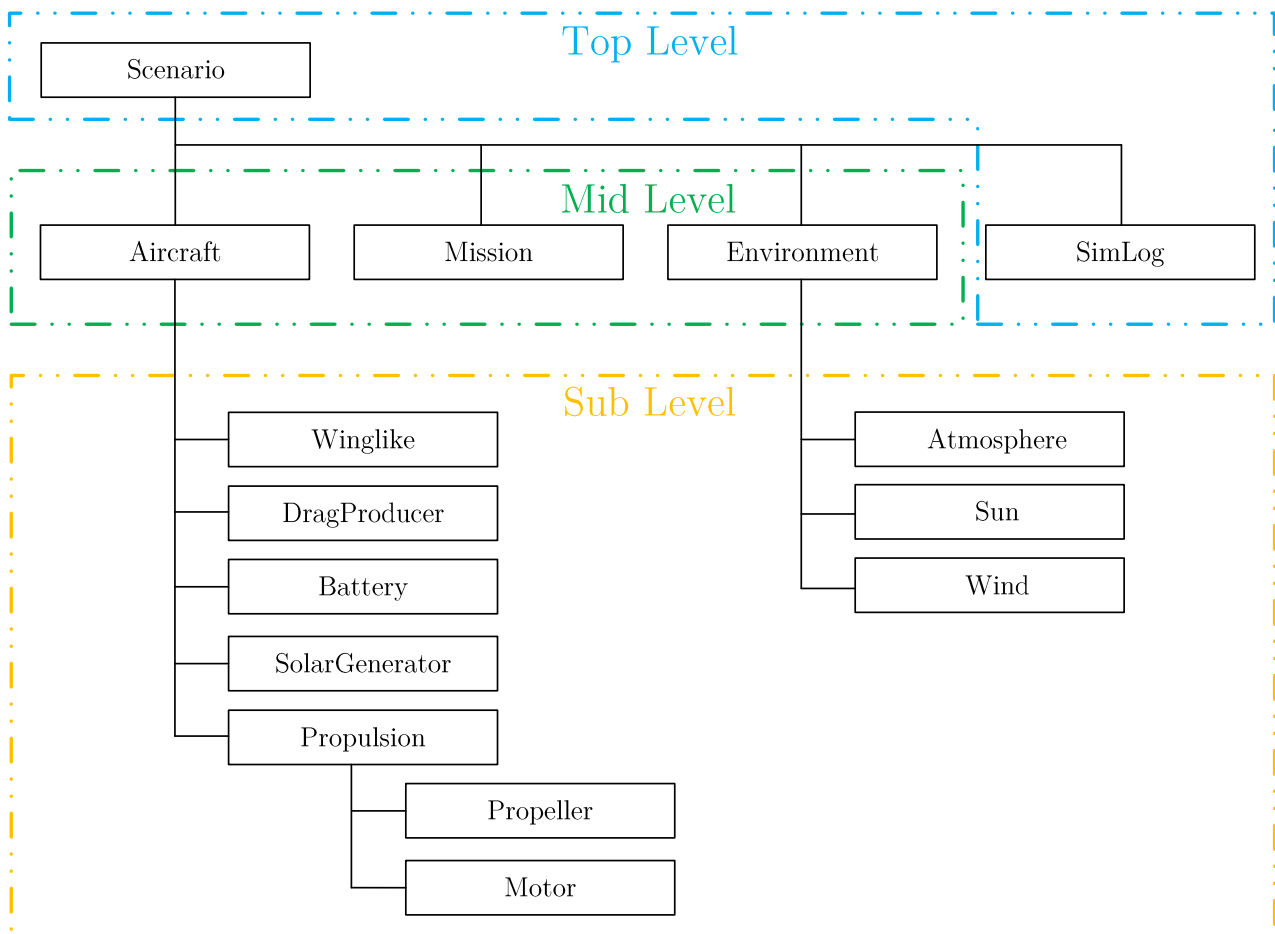


Abbildung 3.2: Gesamte Klassenstruktur

Bei der Instanziierung der Objekte werden die jeweiligen Klassen mit verschiedenen Eingabedaten versorgt. Diese befinden sich zentral in einer separaten Ordnerstruktur bezeichnet mit **UserData**. Dies erhöht die Übersichtlichkeit im Umgang mit dem Programm. Innerhalb dieser Ordnerstruktur sind weitere Eingabedateien und Ordner hinterlegt jeweils den zugehörigen Klassen bezeichnet. Die betreffende Klasse lädt sich jeweils ihre benötigten Daten eigenständig anhand der gewählten Simulation.

3.3 Ablauf der Simulation

Bei Start der Simulation werden im Skript **PerformSimulation** zuerst die gewählten Datenpfade für das zu simulierende Flugzeug, die gewählte Mission und das Windmodell eingelesen. Diese Pfade werden bei der darauffolgenden Instanziierung des Szenario übergeben. Die Funktion **Scenario.simulate** triggert die Hauptsimulation an. Innerhalb dieser Funktion wird eine Schleife ausgeführt, die pro Durchgang einen definierten Zeitschritt simuliert. Die Anzahl der Durchgänge wird stetig überprüft und durch entsprechende Marker begrenzt. Weiterhin werden Funktionen der untergeordneten Mid Level Objekte aufgerufen. Beginnend mit der Funktion **Mission.GetFlightData** werden die Parameter der gewünschten Flugbahn gesetzt. Darauf folgt **Environment.SetData** und setzt sämtliche Parameter der Atmosphäre, des Sonnenstandes und des Windes. Die Funktion **Aircraft.CalculateFlightState** berechnet die möglichen Bahnparameter abhängig vom aktuellen Flugzeugzustand und der Umgebung. Diese werden an **Mission.Optimizer** übergeben um künftige Bahn-

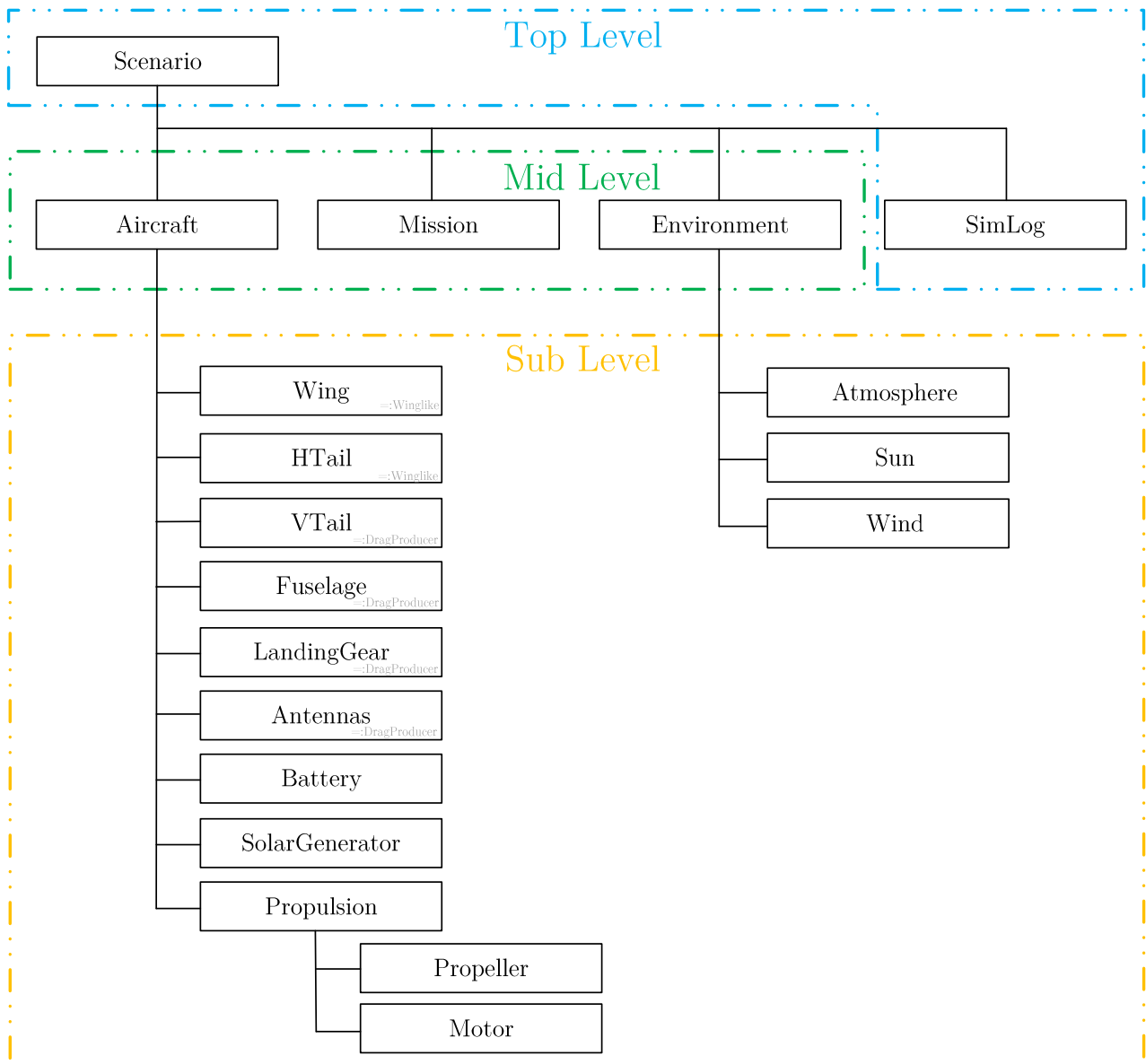


Abbildung 3.3: Instanzierte Objekte und ihre Struktur

planungen zu verbessern. Primär sollen hier zu hohe Werte reduziert werden, um die Simulation zu beschleunigen. Darauf wird der eigentliche Flugschritt mit **Aircraft.ExecuteFlightStep** ausgeführt und sämtliche Werte am Flugzeug verbindlich gesetzt. Innerhalb des Scenario werden neue Daten für die Position des Flugzeuges und der Zeit gesetzt. Am Schleifenende werden alle Daten durch die Funktion **SimLog.LogData** gespeichert zur späteren Analyse. Dann erfolgt der nächste Schleifendurchgang. Der beschriebene Ablauf ist in Abbildung 3.4 dargestellt und listet die zur Simulation notwendigen Hauptfunktionen auf.

Die detaillierte Beschreibung der Funktionen jeder Klasse, sowie ihre Zusammenhänge zu anderen Objekten schließt in den nachfolgenden Kapiteln an. Dies schließt die Funktionen der Sub Level Objekte mit ein.

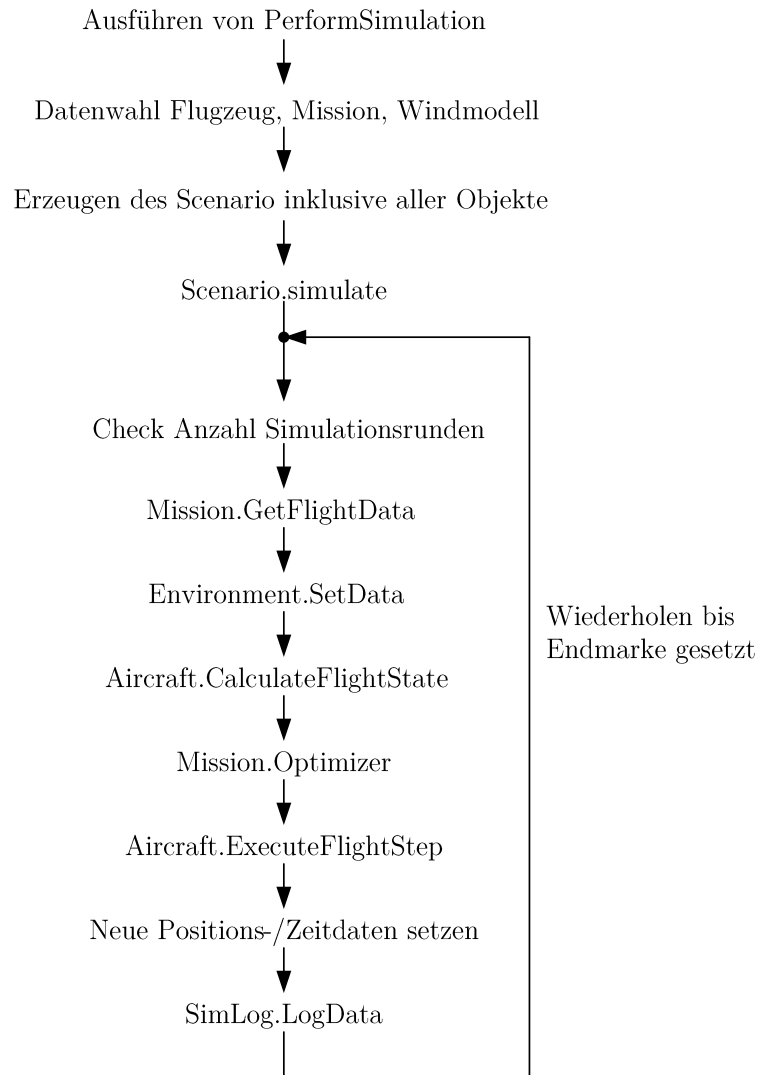


Abbildung 3.4: Simulationsablauf

3.4 Klassen und Funktionen

Im Folgenden sind die Klassen beschrieben. Neben ihren Attributen und Funktionen werden auch Zusammenhänge und Verbindungen zu anderen Klassen aufgezeigt. Klassendiagramme unterstützen die Erläuterungen. Es ist anzumerken, dass klassenintern genutzte Attribute, die nur den Ablauf erleichtern, nicht aufgeführt bzw. erklärt werden. Weiterhin sind aufgeführte Parameter bewusst nicht im Symbolverzeichnis hinterlegt, da die Erklärung des Parameters in der entsprechenden Abbildung geliefert wird. Eine Übersicht aller Parameter ist in Anhang D gegeben.

3.4.1 Top Level

3.4.1.1 Scenario

Das Scenario bildet den Kern der Simulation, weshalb es bereits mehrfach teilweise beschrieben wurde. Das entsprechende Klassendiagramm des Scenario ist in Abbildung 3.5 dargestellt und zeigt rechtsseitig zusätzliche Bezeichnungen und Einheiten der Attribute. Im mittleren Block der Attribute sind neben den schon bekannten Objekten weitere Parameter aufgelistet.

Scenario	Attributbezeichnung	Einheit
Aircraft	Objekt	-
Environment	Objekt	-
Mission	Objekt	-
SimLog	Objekt	-
time	Aktuelle Zeit	[s]
day	Aktueller Tag	[-]
iter	Feld mit Iterationsparametern	-
x_g	X-Koordinate im geodätischen KOS	[m]
y_g	Y-Koordinate im geodätischen KOS	[m]
z_g	Z-Koordinate im geodätischen KOS	[m]
alt	Aktuelle Höhe	[m]
Simulate		
LogData		

Abbildung 3.5: Klasse Scenario inklusive Bezeichnungen

Die Aufnahme der aktuellen Zeit und des Tages dient neben der nachfolgenden Datenanalyse auch zur Berechnung des Sonnenstandes in der Sub Level Klasse Sun. Das Feld **iter** besteht aus einem Datenarray mit Iterationsparametern. Diese sind in Tabelle 3.1 aufgeführt. Der Parameter **mode** von iter unterscheidet zwischen einer regulären Simulation oder einer Testsimulation zur Verifikation von Werten. Die drei folgenden Parameter **dt**, **ds** und **da** sind von besonderer Bedeutung, da sie die Schrittweiten eines Simulationsschrittes bzw. einer Lastreduktion festlegen. **threshold** setzt die Genauigkeit interner Iterationen fest. **counter_max** dient als Überwachungswert der Iterationen. Sollten Werte nicht konvergieren und die maximale Anzahl an Iterationen überschreiten, wird die Simulation mit einer Warnung beendet. Die Warnung gibt an, an welcher Stelle es zu einem Überlauf kam. **overload** ist ein Marker, der gesetzt wird, wenn Lastlimits erreicht werden. Die restlichen Parameter sind selbsterklärend.

Parameter	Bezeichnung	Einheit
mode	Modus der Simulation	
dt	Zeitschritt für Simulation	s
ds	Schrittweite zur Lastreduktion von Geschwindigkeiten	m/s
da	Schrittweite zur Lastreduktion von Winkeln	rad
sim_round	individueller Zähler	-
sim_maxrounds	max. Anzahl an Simulationsrunden	-
threshold	Schwellwert für Iterationen	-
counter_max	max. Anzahl an Iterationen	-
checkpoint	Zeitmarkierung in Flugpfadausgabe	min
delta_0	Startdifferenz	-
counter_0	Startzähler	-
overload	Überlast Indikator	-

Tabelle 3.1: Parameter des Arrays iter

Die Attribute **x_g**, **y_g** und **z_g** entsprechen den Positionsdaten des Flugzeuges im geodätischen Koordinatensystem. Diese Werte werden von dem Objekt Mission verwendet zur weiteren Bahnplanung. Letzter aufgeführter Wert **alt** spiegelt die aktuelle Flughöhe wieder.

Die Funktion **Simulate** wurde im vorangegangenen Kapitel bei der Beschreibung des Gesamtablaufs bereits beschrieben. Der untere Schleifenblock aus Abbildung 3.4 führt alle Schritte dieser Funktion auf. **LogData** dient der Datenspeicherung. In ihr werden in einer Schleife sämtliche Parameter aufgerufen und über eine Funktion des Objekts SimLog gespeichert.

3.4.1.2 SimLog

Die Klasse SimLog dient der Datenspeicherung, -verarbeitung und -ausgabe. Das zugehörige Klassendiagramm ist in Abbildung 3.6 gezeigt. Während der Simulation ist einzig die Funktion **LogValue** aktiv. Mit ihr wird ein Datenwert abgespeichert und entsprechend indiziert. Das Objekt Scenario wiederholt den Aufruf von **SimLog.LogValue** bis sämtliche Parameter gespeichert wurden. Die Parameternamen sowie -adressen sind in **DataNames** hinterlegt. Weiterhin sind hier ausführliche Bezeichnungen und Einheiten für beispielsweise Achsenbeschriftungen hinterlegt. Die eigentlichen Zahlenwerte der Parameter werden in **Data** gespeichert. Die Größe dieses Feldes entspricht der Anzahl der maximal durchgeführten Simulationsrunden. Die übrigen Attribute dienen der internen Verrechnung.

Die Funktion **ListData** speichert sämtliche Werte eines selektierten Parameters in ein Datenarray ab. Damit können ausgewählte Datensätze separat analysiert werden. Die Funktionen beginnend mit **Plot** erzeugen verschiedene Datendiagramme, abweichend in Darstellung und Formatierung. Die letzte Funktion **PlotFlightPath** unterscheidet sich den Vorgängern insofern, dass sie nur den Flugpfad in drei Achsen visualisiert.

SimLog	Attributbezeichnung	Einheit
DataNames	Feld mit Datennamen	[-]
Data	Feld mit Datenwerten	[-]
n_data	Anzahl der zu speichernden Daten	[-]
timestamp_0	Zeitstempel zu Simulationsbeginn	[s]
ListData		
LogValue		
PlotData		
PlotDataOverAlt		
PlotIterlogs		
PlotThreeData		
PlotFlightPath		

Abbildung 3.6: Klasse SimLog inklusive Bezeichnungen

3.4.2 Mid Level

3.4.2.1 Aircraft

Die umfangreichste Klasse bildet das Aircraft. Sie dient der Ermittlung möglicher Flugzustände und -leistungen sowie der Umsetzung dieser. Des Weiteren werden Energien zwischen den einzelnen Energielieferanten und -verbrauchern ausgetauscht. Die Fluglage und -orientierung wird ebenfalls erfasst. Eine Übersicht der Klasse ist in Abbildung 3.7 gegeben.

Aircraft	Attributbezeichnung	Einheit
v_min_sl	min. Fluggeschwindigkeit auf SL	[m/s]
v_max_sl	max. Fluggeschwindigkeit auf SL	[m/s]
max_op_FL	max. Betriebsflughöhe	[m]
m	Gesamtmasse Flugzeug	[kg]
S	Flügelfläche	[m ²]
x_CoG	Schwerpunktlage bezogen auf Flugzeugnase	[m]
Wing	Objekt (Typ Winglike)	-
Htail	Objekt (Typ Winglike)	-
Vtail	Objekt (Typ DragProducer)	-
Fuselage	Objekt (Typ DragProducer)	-
LandingGear	Objekt (Typ DragProducer)	-
Antennas	Objekt (Typ DragProducer)	-
Battery	Objekt	-
SolarGenerator	Objekt	-
Propulsion	Objekt	-
P_systems	elektr. Leistung System	[W]
alpha_Wing	Anstellwinkel Flügel α_{Wing}	[rad]
alpha_Tail	Anstellwinkel Höhenleitwerk α_{Htail}	[rad]
Re_Wing	Reynoldszahl Flügel	[-]
CL	Auftriebskoeffizient	[-]
CD	Widerstandskoeffizient	[-]
v	Fluggeschwindigkeit	[m/s]
v_h	Fluggeschwindigkeit horizontal proj.	[m/s]
v_v	Fluggeschwindigkeit vertikal proj.	[m/s]
r_turn	Kurvenradius des geflogenen Kreises	[m]
theta	Nickwinkel θ	[rad]
gamma	Bahnneigungswinkel γ	[rad]
phi_roll	Rollwinkel ϕ_{roll}	[rad]
psi	Gierwinkel ψ	[rad]
N_props	Anzahl der Propeller	[-]
Iterlog	Feld mit Iterationsmarkern	-
CalculateFlightState		
CheckAirspeedAllowed		
ExecuteFlightStep		
GetLoadReduction		

Abbildung 3.7: Klasse Aircraft inklusive Bezeichnungen

Die ersten drei Attribute **v_min_sl**, **v_max_sl** und **max_op_FL** dienen der Absicherung der Flugenveloppe, also der Flugbereichsgrenzen. Dazu sind minimale und maximale Fluggeschwindigkeiten für Sea Level (SL) angegeben. Die Angaben werden in EAS angegeben. Die maximale Betriebsflughöhe begrenzt das Einsatzspektrum nach oben. Bei korrekten Missionsdaten werden die Grenzen nie erreicht. Die Eigenschaft **m** steht für das Gesamtmasse des Flugzeuges, meist verwendet zur Berechnung des Auftriebes. Darauf folgt die Flügelfläche **S** und die relative Schwerpunktlage **x_CoG** bezogen auf die Rumpfnase. Die danach folgenden Objekte wurden bereits genannt und gehören zu Sub Level Klassen, die im folgenden Unterkapitel genauer beschrieben werden. Das Attribut **P_systems** umfasst die notwendige Leistung zur Versorgung von Bordsystemen. Die Anstellwinkel für den Flügel und das Höhenleitwerk, die Reynoldszahl sowie der Auftriebs- und Widerstandskoeffizient bilden die primären aerodynamischen Parameter. Mit ihnen wird ein Großteil der Flugleistungen bestimmt. Die Fluggeschwindigkeit **v** wird zusätzlich in ihren Komponenten angegeben. Diese entsprechen dem hori-

zontalen Anteil entlang der geodätischen X-Achse und dem vertikalen Anteil entlang der geodätischen Z-Achse. Der Parameter des Kurvenradius des geflogenen Kreises wird genutzt zur Bestimmung des Rollwinkels. Der folgende Nick-, Bahnneigungs-, Roll- und Gierwinkel beschreibt jeweils eine Flugorientierung des Flugzeuges. Die Anzahl der Propeller und folglich Motoren wird durch **N_props** gegeben. Das Feld **Iterlog** enthält mehrere Marker gekennzeichnet für den zugehörigen Programmabschnitt einer Iterationsschleife. Die Marker werden zur Analyse der Simulationsgeschwindigkeit genutzt, da diese die Anzahl an durchlaufenen Iterationsschleifen pro Simulationsschritt an jeweiliger Stelle zählen. Des Weiteren werden sie zur Verifizierung des Programmablaufes verwendet.

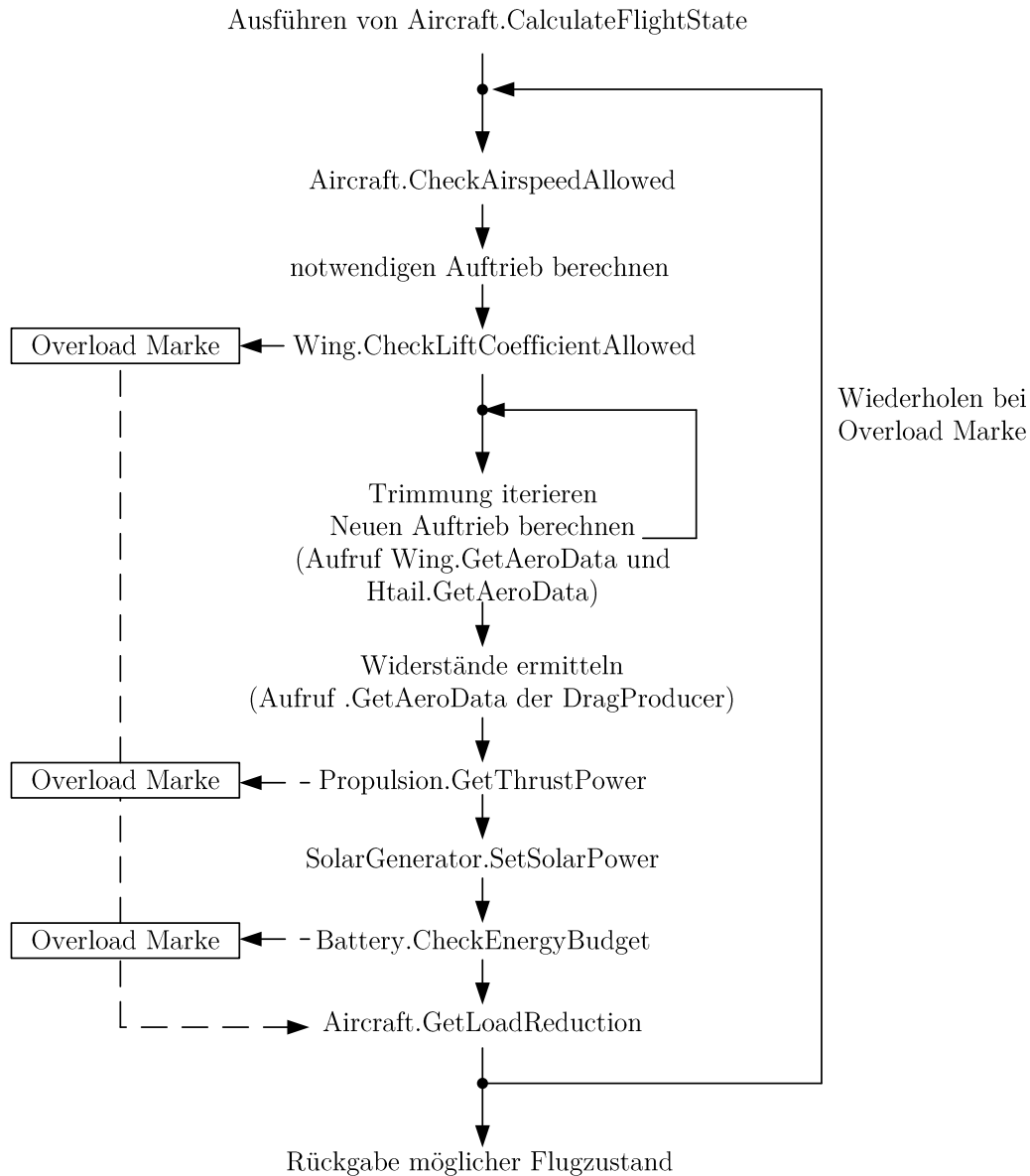


Abbildung 3.8: Ablauf und Aufrufe innerhalb der Funktion CalculateFlightState

Die Funktion **CalculateFlightState** wird ablauftechnisch zuerst abgerufen, welche die vom Flugzeugzustand möglichen Flugleistungen mit den geforderten Flugleistungen aus der Missionsklasse vergleicht. Der Ablauf innerhalb dieser Funktion ist in Abbildung 3.8 dargestellt und zeigt dort die Verknüpfungen zu Sub Level Klassen. Darauf folgt die Funktion **CheckAirspeedAllowed**.

Diese prüft, ob die geforderten Werte für Fluggeschwindigkeit und/oder -höhe innerhalb der Flugbereichsgrenzen liegen. Wie zuvor genannt sollte diese Funktion nie anschlagen bei korrekten Missionsdaten. Weiterhin greift die Funktion **GetLoadReduction** innerhalb dieses Ablaufes. Diese reduziert die geforderte Flugleistung. Als primäre Steuergröße wird der Bahnneigungswinkel γ verwendet. Es ist anzumerken, dass hier noch keine Daten gesetzt bzw. umgesetzt werden. Erst die Funktion **ExecuteFlightStep** setzt alle relevanten Parameter bindend. Neben der Berechnung von Positions- und Leistungsdaten wird hier die Funktion `Battery.ExchangeEnergy` aufgerufen.

3.4.2.2 Mission

Mission	Attributbezeichnung	Einheit
timestamp_0	Startuhrzeit im 24h System	[s]
daystamp_0	Starttag bezogen auf 365 Tage im Jahr	[-]
phi	Breitengrad ϕ	[rad]
lambda	Längengrad λ	[rad]
section	Missionsabschnitt	-
section_type	Art des Missionsabschnittes	-
S_names	Feld mit Abschnittsnamen	-
S_data	Feld mit Abschnittsdaten	-
delta	Differenzparameter	[-]
GetFlightData		
Optimizer		

Abbildung 3.9: Klasse Mission inklusive Bezeichnungen

Die Steuerung des Flugpfades wird durch die Klasse **Mission** umgesetzt. Sie liefert weiterhin die Standort- und Zeitdaten für den Simulationsstart und unterscheidet die geplante Mission in Abschnitte. Im Klassendiagramm in Abbildung 3.9 ist das Attribut für die Startzeit **timestamp_0** im 24 *h*-System in Sekunden angegeben. Der Starttag **daystamp_0** beginnt am Jahresanfang bei 0 und kann bis zum Tag 365 gesetzt werden. Der Breitengrad **phi** und Längengrad **lambda** ergibt die Startposition auf der Erde angegeben in rad. Die vier genannten Parameter werden von der Sub Level Klasse `Sun` benutzt. Die folgenden Eigenschaften zu den Missionsabschnitten enthalten teilweise weitere Felder. **S_names** enthält die zu **S_data** zugehörigen Bezeichnungen und Einheiten der Daten. In **S_data** ist ein Array hinterlegt mit Bahnparametern für den jeweiligen Abschnitt chronologisch aufsteigend sortiert. Ein Beispiel dazu ist in Tabelle 3.2 gezeigt.

Wert	Bezeichnung	Einheit
0,1	Bahnneigungswinkel γ	[rad]
5000	Radius der geflogenen Kreisbahn	[m]
9	Fluggeschwindigkeit v in EAS	[m/s]
20000	Zielhöhe	[m]
0	Zielzeit	[h]
0	Priorität	[-]
	0:= Höhe, 1:=zeit	

Tabelle 3.2: Beispieldaten eines Abschnittes

Die Abschnitte lassen sich grob unterteilen abhängig von der geforderten Flugbahn. Tabelle 3.3 zeigt die möglichen Abschnittsarten. Grob lässt sich unterteilen in Steigflug, Reiseflug und Sinkflug. Eine beliebige Kombination und Aneinanderreihung gleicher und/oder unterschiedlicher Abschnittsarten mit unterschiedlichen Bahnparametern ist beliebig möglich. Dies ermöglicht beispielsweise eine Unterbrechung des Steigfluges auf Zwischenhöhen um Ladephasen der Batterien zu erzeugen. Weiterhin kann nach dem Abschnittsziel unterschieden werden. Entweder soll eine definierte Zielhöhe erreicht werden oder der Abschnitt wird nach Ablauf einer Zeit beendet. Stehen keine Abschnitte vor Ablauf der maximalen Simulationsschritte mehr an, wird die Simulation vorzeitig beendet. An den User erfolgt dann die Rückmeldung der erfolgreich durchgeführten Mission.

Art	Richtung	Abschnittsziel
Steigflug	links gekrümmt	Höhe/Zeit
”	gerade	”
”	rechts gekrümmt	”
Reiseflug	links gekrümmt	Zeit
”	gerade	”
”	rechts gekrümmt	”
Sinkflug	links gekrümmt	Höhe/Zeit
”	gerade	”
”	rechts gekrümmt	”

Tabelle 3.3: Abschnittsarten

Die Funktion **GetFlightData** liefert die notwendigen Bahnparameter als Rückgabe an das Scenario abhängig vom aktuellen Missionsabschnitt. Es werden Angaben zum Bahnneigungswinkel γ , dem Radius des zu fliegenden Flugkreises r_{turn} sowie die geforderte Fluggeschwindigkeit v in EAS bereitgestellt. Das noch nicht genannte Attribut **delta** ist ein Optimierungsparameter für die Bahnplanung. Es wird primär von der Funktion **Optimizer** verwendet. Diese vergleicht die zuvor geforderten Bahnparameter mit den aktuell vom Flugzeug erreichbaren. Kann das Flugzeug durch seinen aktuellen Zustand die Bahnparameter nicht erfüllen und befindet sich der Folgeschritt im gleichen Missionsabschnitt werden reduzierte Werte nachfolgend angefordert. Dies reduziert hauptsächlich interne Iterationsdurchläufe und spart Simulationszeit. Zum Zeitpunkt der Umsetzung konnten so ca. 37% der Simulationszeit reduziert werden.

3.4.2.3 Environment

Die Klasse **Environment**, dargestellt in Abbildung 3.10 beinhaltet die Sub Level Klassen Atmosphere, Sun und Wind. Sie selbst dient lediglich der Untergliederung der untergeordneten Klassen. Daher enthält sie keine eigenen Funktionen.

Environment	Attributbezeichnung	Einheit
Atmosphäre	Objekt	-
Sun	Objekt	-
Wind	Objekt	-

Abbildung 3.10: Klasse Environment

3.4.3 Sub Level

3.4.3.1 Winglike

Mit der Klasse Winglike werden hauptsächlich Auftriebs- und Widerstandskoeffizienten des jeweiligen Objektes bestimmt und ausgegeben. Es wird in Abbildung 3.11 dargestellt. Hinter dem Attribut **Name** steht der Name des verwendeten Flügelprofils. In den zugehörigen Kennfeldern **Data** befinden sich mehrdimensionale Datensätze, die Angaben zu Auftriebs-, Widerstands- und Momentenbeiwert beinhalten für verschiedene Reynoldszahlen abhängig vom Anstellwinkel des Flügels. Auf diese greifen die klasseneigenen Funktionen zu. Es ist anzumerken, dass die hinterlegten Daten nur für das Flügelprofil gelten. Die restlichen Attribute werden lediglich in den Berechnungen der Funktionen verwendet. Hier ist das Attribut **CD0_fac** zu erläutern. Es handelt sich um einen Proportionalitätsfaktor für den Widerstandskoeffizienten aufgrund der applizierten Solarpaneele. Dieser Wert wurde durch die Aerodynamikgruppe des Projektes bestimmt.

Winglike	Attributbezeichnung	Einheit
Name	Name des Flügelprofils	-
Data	Kennfelder	-
Re	Reynoldszahl	[-]
alpha	Anstellwinkel	[rad]
S	Flügelfläche	[m ²]
S_wet	Bespülte Oberfläche	[m ²]
c	Flügeltiefe	[m]
lambda	Streckung	[-]
eff	Oswaldfaktor	[-]
CD0_fac	Aufschlagsfaktor wegen Solarpaneele	[-]
x_AC	Position aerodynamische Zentrum	[m]
GetAeroData	bezogen auf Flugzeugnase	
CheckLiftCoeffAllowed		

Abbildung 3.11: Klasse Winglike

Die Funktion **GetAeroData** bestimmt die zum geforderten Auftriebskoeffizienten zugehörigen Daten zu Widerstands- und Momentenkoeffizient, sowie den Anstellwinkel des Flügels. Die Werte werden aus den Kennfeldern interpoliert. Gleiches gilt für die Funktion **CheckLiftCoeffAllowed**. Diese prüft ab, ob der übergebene Auftriebskoeffizient die Grenzen der Polarendaten überschreitet. Werden die Flugbereichsgrenzen eingehalten, wird diese Funktion nie anschlagen.

3.4.3.2 DragProducer

Die Klasse DragProducer bildet zusätzliche Komponenten am Flugzeug ab, die keinen Beitrag am Auftrieb leisten, jedoch einen Widerstand erzeugen. Beispiele dafür sind Antennenanbauten, der Rumpf und das Fahrwerk. Hinter den Attributen in Abbildung 3.12 sind vorhandene Kennwerte der entsprechenden Bauteile hinterlegt. Im aktuellen Programmstand weisen sämtliche Objekte dieser Klasse keinen Auftriebskoeffizienten auf bzw. der Wert Null ist hinterlegt. Die Funktion **GetAeroData** berechnet die Reynoldszahl und gibt diese mit dem Widerstandskoeffizienten zurück an die anfordernde Klasse.

DragProducer	Attributbezeichnung	Einheit
CL	Auftriebskoeffizient	[-]
CD	Widerkoeffizient	[-]
S	Bespülte Oberfläche	[m ²]
lref	Referenzlänge	[m]
GetAeroData		

Abbildung 3.12: Klasse DragProducer

3.4.3.3 Battery

Der zentrale Knotenpunkt für den Energieaustausch bildet die Klasse Battery, dargestellt in Abbildung 3.13. Sämtliche Energieflüsse, sowie das Batteriemangement laufen hier zusammen. Neben dem Attribut der Batteriemasse **m** ist die Energiedichte **e** der Batterien einer der wichtigsten Parameter. Letzteres wird noch durch **e_0** und **r_eTT25** skaliert zur Abbildung einer Temperaturabhängigkeit.

Battery	Attributbezeichnung	Einheit
m	Masse der Batterien	[kg]
e	Energiedichte	[Ws/kg]
e_0	Theor. Energiedichte	[Ws/kg]
r_eTT25	Temperaturabhängigkeit Energiedichten	[-]
E	Eigentliche elektr. Energie	[Ws]
E_max	Max. elektr. Energie	[Ws]
E_charge	Geladene elektr. Energie	[Ws]
E_discharge	Entladene elektr. Energie	[Ws]
E_heat	Heizenergie	[Ws]
SoC	Ladestand der Batterien	[-]
SoC_min	Minimaler Ladestand der Batterien	[-]
eta_chg	Wirkungsgrad Laden	[-]
eta_dsg	Wirkungsgrad Entladen	[-]
zeta_dgr	Degradierungsfaktor	[-]
zeta_dgr_0	Degradierungsfaktor bei Start	[-]
zeta_dgr_n	Degradierungsfaktor nach Zyklen	[-]
n_chg	Äquivalenter Ladezyklus	[-]
T_heat	Zu heizende Batterietemperatur	[K]
HeatMap	Kennfeld zur Heizleistung	-
Charge		
CheckEnergyBudget		
Discharge		
ExchangeEnergy		
InitHeating		
SetHeating		

Abbildung 3.13: Klasse Battery

Die darauf folgenden Energien sind unterteilt nach ihrem Zweck. Die Eigenschaft **SoC** steht für State of Charge und gibt den Ladestand der Batterien wieder. Sinkt dieser unter das Minimalniveau **SoC_min** gilt die Mission als gescheitert. Die Simulation wird dennoch weiter geführt zur Datenanalyse. Jedoch mit konstant angegebenen minimalen Ladestand. Die Wirkungsgrade sind selbsterklärend. Der Degra-

dierungsfaktor **zeta_dgr** ist abhängig von der Anfangsdegradierung und der Anzahl an Ladezyklen, jeweils ausgedrückt durch **zeta_dgr_0** und **zeta_dgr_n**. Zu dieser Berechnung gehört auch der äquivalente Ladezyklus **n_chg**. Das Attribut **T_heat** fließt in nachfolgende Eigenschaft mit ein zur Bestimmung der Heizleistung. Im Kennfeld **HeatMap** sind diese Leistungen für bestimmte Batterietemperaturen abhängig von der Flughöhe hinterlegt.

Bei Simulationsstart wird durch **InitHeating** die Energiedichte durch die Temperatur entsprechend skaliert. Die Funktion **CheckEnergyBudget** prüft ab, ob die für den jeweiligen Simulationsschritt geforderte Energiemenge durch den Restladestand der Batterie abgebildet werden kann. In ihr wird auch die Funktion **SetHeating** verwendet. Diese berechnet die notwendige Heizleistung für die Batterien. Durch **ExchangeEnergy** werden die jeweiligen Energien ausgetauscht und je nach Energieüberschuss oder -mangel durch die Funktionen **Charge** und **Discharge** umgesetzt.

3.4.3.4 SolarGenerator

Die Klasse SolarGenerator dient der Umsetzung der solaren Strahlung in nutzbare Energie. Das Attribut **SolarPanels** enthält die Geometrie- und Orientierungsdaten der einzelnen Solarzellen. Individuelle Neigungen durch den Flugzeugflügel sind dadurch berücksichtigt. Die nachfolgenden Attribute werden zur Berechnung der gesamten Energie durch den Solargenerator verwendet. Hier ist die Transformationsmatrix **T_g_p** zu erläutern, die aus einer Zusammensetzung aus Drehmatrizen besteht. Mir ihr kann vom geodätischen Koordinatensystem in das jedes einzelnen Panels transformiert werden. Dies wird mitunter in der Funktion **SetSolarPower** genutzt. Abbildung 3.14 zeigt das Klassendiagramm des SolarGenerators.

SolarGenerator	Attributbezeichnung	Einheit
SolarPanels	Feld mit Daten zu Solarpanelen	-
N_Panels	Anzahl der Solarpanele	[-]
A_SG	Fläche des Solargenerators	[m ²]
Eff_SG	Wirkungsgrad des Solargenerators	[-]
P_SG	Energie durch Solargenerator	[W]
p_SG	Spez. Energie durch Solargenerator	[W/m ²]
T_g_p	Transformationsmatrix	-
SetSolarPower	geodätisch nach panel	

Abbildung 3.14: Klasse SolarGenerator

3.4.3.5 Propulsion

Die Propulsionklasse bildet den gesamten Antriebsstrang des Flugzeuges ab. Dazu gehören die untergeordneten Objekte **Motor** und **Propeller** wie in Abbildung 3.15 gezeigt. Der Motor steht sinnbildlich für den Elektromotor eines Antriebes inklusive der Abtriebswelle. Der Propeller wiederum steht für den Propeller mit der verbundenen Abtriebswelle, jedoch propellerseitig betrachtet. Neben der gesamten elektrischen Leistung **P_el_tot** ist noch die Anzahl der Antriebsstränge **N_Props** angegeben als Lesedaten für die Klasse Battery. Das Attribut **Iterlog** dient wie alle Iterlogs der programminternen Verfolgung der Anzahl der Iterationen.

Propulsion	Attributbezeichnung	Einheit
Motor	Objekt	-
Propeller	Objekt	-
P_el_tot	Elektrische Leistung	[W]
N_Props	Anzahl der Propeller	[-]
Iterlog	Feld mit Iterationsmarkern	-
GetThrustPower		

Abbildung 3.15: Klasse Propulsion

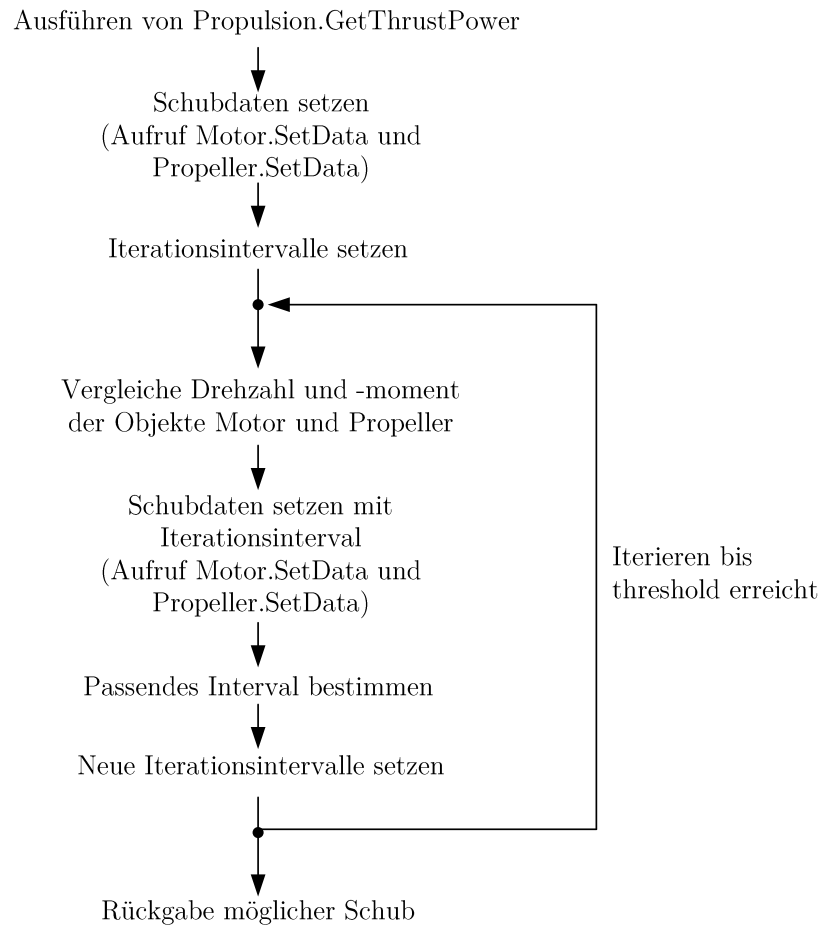


Abbildung 3.16: Ablauf und Aufrufe innerhalb der Funktion GetThrustPower

Die Funktion **GetThrustPower** vergleicht den geforderten Schub mit dem aktuell möglichen Schub. Zur Ermittlung des Schubes werden Kennlinien der Objekte Motor und Propeller iteriert bis ein gemeinsamer Schnittpunkt gefunden ist. Befindet sich die geforderte Leistung bereits innerhalb der Kennfeldes des Motors wird der entsprechende Betriebspunkt gesetzt. Der Ablauf der Funktion ist in Abbildung 3.16 verdeutlicht.

3.4.3.6 Propeller

Die Klasse Propeller, dargestellt in Abbildung 3.17, setzt verschiedene Schubparameter abhängig von Leistungsparametern. Diese Leistungsparameter sind in den Kennfeldern des Fortschrittgrades, des Schubkoeffizienten und des Leistungskoeffizienten hinterlegt.

Abhängig von der Drehzahl und der Umgebung setzt die Funktion **SetData** mögliche Werte für den Propellerschub bzw. die Propellerleistung. Die Interpolation innerhalb der Kennfelder legt weiterhin minimale und maximale Werte für den Schub fest.

Propeller	Attributbezeichnung	Einheit
D_p	Propellerdurchmesser	[m]
J	Kennfeld Fortschrittsgrad	[-]
CT	Kennfeld Schubkoeffizient	[-]
CP	Kennfeld Leistungskoeffizient	[-]
T_prop	Schub Propeller (gesamt)	[N]
P_prop_aero	Aerodyn. Leistung Propeller	[W]
Eta_prop	Wirkungsgrad Propeller	[-]
N_prop	Drehzahl Propellerwelle	[1/s]
P_prop	Leistung Propellerwelle	[W]
M_prop	Drehmoment Propellerwelle	[Nm]
T_min	Minimaler Schub	[N]
T_max	Maximaler Schub	[N]
SetData		

Abbildung 3.17: Klasse Propeller

3.4.3.7 Motor

Die Klasse Motor ist das Pendant zur Propellerklasse. Die Funktion **SetData** bestimmt abhängig von der Motorkennlinie, die im Attribut **Parameters** hinterlegt ist, die Leistungswerte des Motors. Dazu gehören die elektrische Leistung sowie das Drehmoment und die Drehzahl der Abtriebswelle. Das Klassendiagramm des Motors ist in Abbildung 3.18 dargestellt.

Motor	Attributbezeichnung	Einheit
Parameters	Motorkenndaten zum Einlesen	-
Eta_el	Elektrische Wirkungsgrad	[-]
P_el	Elektrische Leistung	[W]
P_shaft	Leistung der Abtriebswelle	[W]
M_shaft	Drehmoment der Abtriebswelle	[Nm]
N_shaft	Drehzahl der Abtriebswelle	[1/s]
SetData		

Abbildung 3.18: Klasse Motor

3.4.3.8 Atmosphäre

Die Klasse `Atmosphäre` stellt Atmosphären­daten bereit und greift dazu primär auf ein Kennfeld mit Daten der Standard Atmosphäre zu. Das entsprechende Klassendiagramm ist in Abbildung 3.19 auf­geführt. Diese Daten sind im Attribut **Atmo** hinterlegt. Weitere Attribute sind die Dichte, die dy­namische Viskosität und die kinematische Zähigkeit jeweils bezogen auf Meereshöhe verwendet durch Funktionen der Klasse `Aircraft`. Die restlichen Eigenschaften spiegeln die im aktuellen Simulations­schritt verwendeten Luftdaten wieder und werden von der Klasse `Aircraft` und `Sun` benutzt. Die Funktion **SetAtmosphäreData** interpoliert die aktuellen Luftdaten aus denen des Kennfeldes für die aktuelle Flughöhe.

Atmosphäre	Attributbezeichnung	Einheit
Atmo	Kennfeld mit Atmosphären­daten	-
Rho_SL	Dichte auf SL	[kg/m ³]
Eta_SL	Dynamische Viskosität auf SL	[μPas]
Nu_SL	Kinematische Viskosität auf SL	[m ² /s]
T	Temperatur	[K]
P	Druck	[Pa]
Rho	Dichte	[kg/m ³]
Nu	Kinematische Viskosität	[m ² /s]
A	Schallgeschwindigkeit	[m/s]
SetAtmosphäreData		

Abbildung 3.19: Klasse `Atmosphäre`

3.4.3.9 Sun

Zur Berechnung des Sonnenstandes wird die Klasse `Sun` verwendet. Abbildung 3.20 zeigt das verwen­dete Klassendiagramm. Das Attribut **Eps_0** entspricht einer Solarkonstante und wird zur Berechnung der direkten solaren Strahlung **Eps_dir** verwendet. Die weiter aufgeführten Attribute bis zum Er­dradius dienen ebenfalls der Berechnung der direkten solaren Strahlung. Details zu den Rechnungen sind in Kapitel 4 beschrieben. Die Transformationsmatrix **T_g_e** enthält eine Kombination mehrerer Drehmatrizen zur Transformation vom erdfesten zum geodätischen Koordinatensystem. Die Funktion **SetSunData** berechnet sonnenbezogene Kennwerte abhängig von Position und Zeit.

Sun	Attributbezeichnung	Einheit
Eps_0	Solarkonstante	[W/m ²]
Eps_dir	Direkte solare Strahlung	[W/m ²]
Zeta_S	Sonneneinfallswinkel	[rad]
Alpha_S	Höhenwinkel	[rad]
Psi_S	Azimut der Sonne	[rad]
Delta	Deklination der Sonne	[rad]
Tau	Stundenwinkel	[rad]
Zeta_dep	Depression	[rad]
R_earth	Erdradius	[m]
T_g_e	Transformationsmatrix	-
SetSunData	erdfest nach geodätisch	

Abbildung 3.20: Klasse Sun

3.4.3.10 Wind

Die Klasse Wind, dargestellt in Abbildung 3.21, umfasst ein Kennfeld zu Winddaten. Diese Daten sind im Attribut **Windmap** hinterlegt und unterteilen sich weiter in Höhe, Windgeschwindigkeit und Richtung aus die der Wind weht. Die verwendeten Datensätze werden durch den Benutzer vor Simulationsbeginn festgelegt. Die eigentlichen Eigenschaften der Windgeschwindigkeit und -richtung dienen der Verrechnung und Speicherung der für den Simulationsschritt gültigen Winddaten. Die Funktion **SetWindData** interpoliert anhand der aktuellen Flughöhe entnommen aus den Positionsdaten des Szenario die entsprechenden Werte.

Wind	Attributbezeichnung	Einheit
Windmap	Feld mit Winddaten	-
v_w	Windgeschwindigkeit	[m/s]
psi_w	Windrichtung	[rad]
SetWindData		

Abbildung 3.21: Klasse Wind

4 Modellierung des Flugzeuges HAP und seiner Umgebung

Zur Durchführung der Simulation ist es notwendig das Flugzeug sowie seine Umgebung mathematisch zu beschreiben. Diese Modellierung wird im Folgenden aufgeführt, beginnend mit der Umgebung, die aufgeteilt wird in die Atmosphäre, das Sonnenmodell und die Modellierung des Windes. Darauf folgt die Abbildung des Flugzeuges und seiner Komponenten.

4.1 Umgebung

4.1.1 Atmosphäre

Die Atmosphäre wird mit der U.S. Standard Atmosphäre von 1976 [30] modelliert. Die Datensätze von [31] liefern Werte für die normierte Luftdichte, die Temperatur, den Luftdruck, die kinematische Viskosität sowie für die Schallgeschwindigkeit abhängig von der Höhe. Die Schrittweite der Höhenwerte beträgt 100 m. Diese Daten stimmen mit der Normatmosphäre nach DIN ISO 2533 bis 20 km Höhe überein [32]. Sie wurde durch die International Civil Aviation Organization (ICAO) 1975 festgelegt und hat ihre Gültigkeit für nördliche Breiten von 40 °.

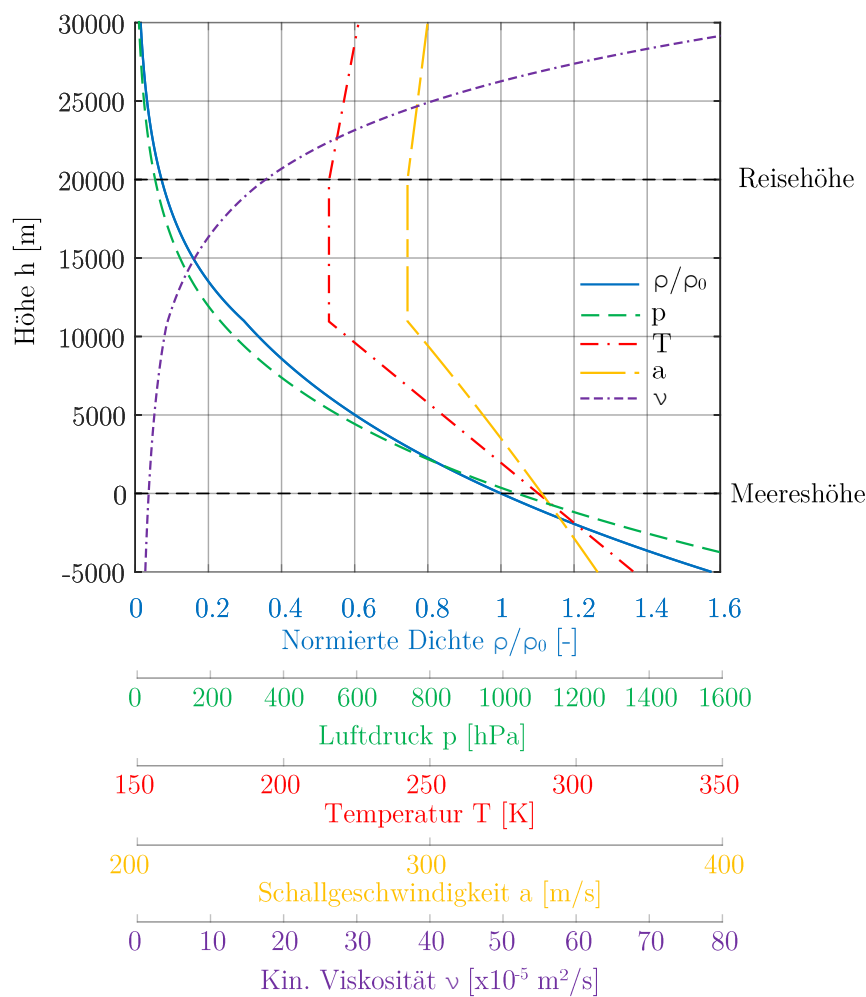


Abbildung 4.1: Atmosphärenparameter verteilt über die Höhe

Die Komponenten der Atmosphäre sind in Abbildung 4.1 aufgeführt. Um Rechenzeiten zu verkürzen wurden die Werte auf 10 m Höhengritte interpoliert und hinterlegt. Die Höhe ist im Bereich von -5.000 bis 30.000 m angegeben. Der negative Höhenbereich dient zur Abbildung von Winterwetterlagen. In der Luftfahrt wird diesbezüglich unter Piloten der Begriff der Dichtehöhe verwendet.

Die aktuelle Dichte $\rho(h)$ einer Höhe kann mit der normierten Dichte ρ/ρ_0 nach Formel 4.1 berechnet werden. Die Dichte auf Meereshöhe ρ_0 kann den Werten aus Tabelle 4.1 entnommen werden.

$$\rho(h) = \frac{\rho}{\rho_0}(h) \cdot \rho_0 \quad (4.1)$$

mit: $\rho(h)$: Luftdichte [kg/m^3]
 $\frac{\rho}{\rho_0}(h)$: normierte Luftdichte abhängig von der Höhe [-]
 ρ_0 : Luftdichte auf Meereshöhe [kg/m^3]

Parameter	Kürzel	Wert	Einheit
rel. Feuchte	f_0	0	%
Druck	p_0	1013,25	hPa
Luftdichte	ρ_0	1,225	kg/m^3
Temperatur	T_0	288,15	K
Schallgeschwindigkeit	a_0	340	m/s
Erdbeschleunigung	g_0	9,80665	m/s^2

Tabelle 4.1: Atmosphärendaten auf Meereshöhe [32]

Ein weitere Komponente, die der Atmosphäre zugeordnet werden kann, ist der Erdradius. Es wird der mittlere Erdradius nach [33] verwendet und beträgt $R_0 = 6.371.000,785$ m. Damit kann die Abnahme der Erdbeschleunigung g mit Formel 4.2 nach [34] modelliert werden. Bei einer angepeilten Reishöhe von 20.000 m nimmt die Erdbeschleunigung um 0,62 % ab.

$$g(h) = g_0 \cdot \left(\frac{R_0}{R_0 + h} \right)^2 \quad (4.2)$$

mit: $g(h)$: Erdbeschleunigung [m/s^2]
 $g_0 = 9,80665 \text{ m/s}^2$: Erdbeschleunigung auf Meereshöhe
 $R_0 = 6.371.000,785 \text{ m}$: mittlerer Erdradius
 h : Flughöhe [m]

4.1.2 Sonnenmodell

Die Modellierung des Solarmodells ist von besonderer Bedeutung, da das zu simulierende Flugzeug solarelektrisch betrieben wird. Die benötigte Energie für den Flug wird durch die auf der flügeloberseite platzierten Solarzellen durch Umwandlung der solaren Strahlung erzeugt. Die Größe und Richtung dieser wird im Folgenden erläutert.

Die solare Strahlung ist abhängig von verschiedensten Faktoren. In der Literatur werden meist empirisch ermittelte Modelle genutzt. Die Basis des hier beschriebenen Solarmodells entspricht [34].

Dabei wird die Strahlung nach der Art des zurückgelegten Weges charakterisiert. Man unterscheidet direkte, diffuse und reflektierte Strahlung. Die direkte Strahlung bezeichnet den Teil der Strahlung, der auf direktem Weg von der Quelle auf den Kontaktpunkt trifft. Die diffuse Strahlung entsteht durch Streuung bzw. Absorption an vorhandenen Partikeln. Sie entsteht an der Erdoberfläche als auch an Partikeln der Atmosphäre, beispielsweise an Wasserdampftröpfchen in Dunst oder Wolken. Die reflektierte Strahlung entsteht durch Reflektion des auftreffenden Lichtstrahls an Oberflächen. [35] [36]

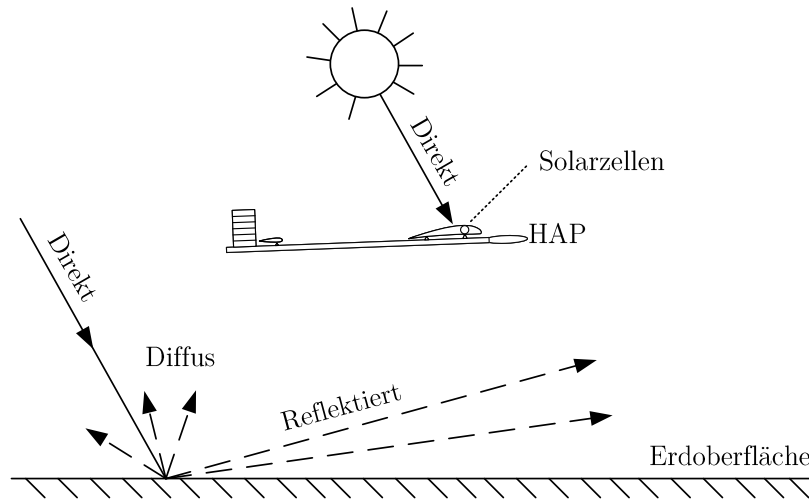


Abbildung 4.2: Arten der solaren Strahlung nach [34]

Abbildung 4.2 zeigt eine schematische Darstellung der vorhandenen Strahlungsarten. Der Anteil der diffusen Strahlung bei geringer Feuchte, wolkenfreier Atmosphäre beträgt ca. 8 % der direkten Strahlung im Bereich der Troposphäre [34]. Aufgrund der Einsatzhöhe des Flugzeuges, sowie der Platzierung der Solarzellen am Flügel wird die diffuse und reflektierte Strahlung vernachlässigt.

Strahlungsmodell

Die direkte Sonnenstrahlung wird mit Formel 4.3 nach [34] berechnet. Sie hat ihre Gültigkeit für große Flughöhen und flache Einstrahlungswinkel der Sonne. Signifikante Abweichungen durch Wolken und andere Wetterphänomene sind hier nicht berücksichtigt, da diese nur schwer analytisch zu modellieren sind. Weiterhin sind diese stark tagesabhängig, was ihre Vorsehbarkeit erschwert. (Anmerkung: Formelparameter in 4.3, die separat berechnet sind, werden anschließend beschrieben)

$$\varepsilon_{dir} = \varepsilon_0(n) \cdot \exp \left(- \frac{c \cdot \exp \left(- \frac{h}{h_s} \right)}{\left[\sin \left(\frac{0,5 \cdot \pi \cdot (\xi + \xi_{dep})}{0,5 \cdot \pi + \xi_{dep}} \right) \right]^{s + \frac{h}{b}}} \right) \quad (4.3)$$

mit:	ε_{dir}	: direkte Sonnenstrahlung [W/m^2]
	$\varepsilon_0(n)$: mittlere Solarkonstante [W/m^2]
	h	: Flughöhe [km]
	$h_s = 7 km$: Skalenhöhe
	ξ	: Zenitwinkel der Sonne / Sonneneinfallswinkel [rad]
	ξ_{dep}	: Depression der Sonne [rad]
	$c = 0,357$: Konstante
	$s = 0,678$: Konstante
	$b = 40 km$: Konstante

Die mittlere Solarkonstante $\varepsilon_0(n)$ kann mit Formel 4.4 berechnet werden. Sie variiert durch den sich ändernden Abstand Erde-Sonne aufgrund der elliptischen Flugbahn der Erde. Die Abweichung beträgt ca. 3 % der mittleren Solarkonstante ε_0 für die ein Wert von $1367 W/m^2$ nach [34] verwendet wird.

$$\varepsilon_0(n) = \varepsilon_0 \cdot \left[1 + 0,033 \cdot \cos \left(\frac{2 \cdot \pi}{365} \cdot (n + 10) \right) \right] \quad (4.4)$$

mit:	$\varepsilon_0(n)$: mittlere Solarkonstante [W/m^2]
	$\varepsilon_0 = 1367 W/m^2$: mittlere Solarkonstante
	n	: Tag im Jahr [-]

Ein weiterer Faktor wird durch die Depression der Sonne ξ_{dep} ausgedrückt. Durch die Beugung des Lichtes innerhalb der Atmosphäre ist die Sonne noch sichtbar, obwohl sie bereits unter dem realen Horizont verschwunden ist. Dieses Phänomen steigt mit der Flughöhe an und kann mit Formel 4.5 beschrieben werden. Dadurch verlängert sich die nutzbare Sonnenzeit.

$$\xi_{dep} = 0,57^\circ + \arccos \left(\frac{R_0}{R_0 + h} \right) \quad (4.5)$$

mit:	ξ_{dep}	: Depression der Sonne [rad]
	$R_0 = 6.371.000,785 m$: mittlerer Erdradius
	h	: Flughöhe [m]

Bestimmung des Sonneneinfalls

Der Sonneneinfallswinkel ξ , auch als Zenitwinkel bezeichnet, definiert sich als Winkel zwischen dem einfallenden Sonnenstrahl und der Normalen zur Erdoberfläche, dargestellt in Abbildung 4.3. Weiterhin ist darin der Höhenwinkel α_S dargestellt. Darüber hinaus ist die Orientierung des Zenitwinkels bezüglich der Ausrichtung Richtung Norden durch den Azimut Ψ_S bestimmt. Dies ist in Abbildung 4.4 im geodätischen Koordinatensystem gezeigt. Der Azimut liegt in der x_g - y_g -Ebene. Normal zu dieser Ebene orientiert durch den Azimut spannt sich die Ebene für den Zenitwinkel auf.

Neben empirischen Formeln zur Berechnung von Zusatzparametern kann der Zenit- und Azimutwinkel rein geometrisch mithilfe von Koordinatentransformationen für beliebige Positionen auf der Erde bestimmt werden.

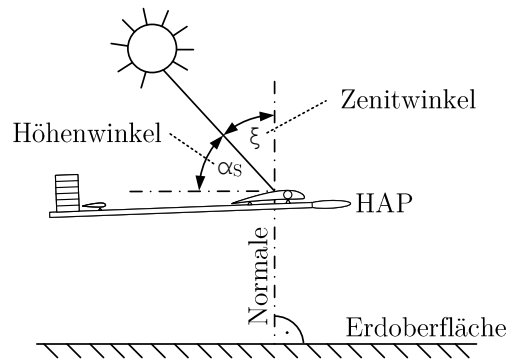


Abbildung 4.3: Zenitwinkel der Sonne in der Ebene

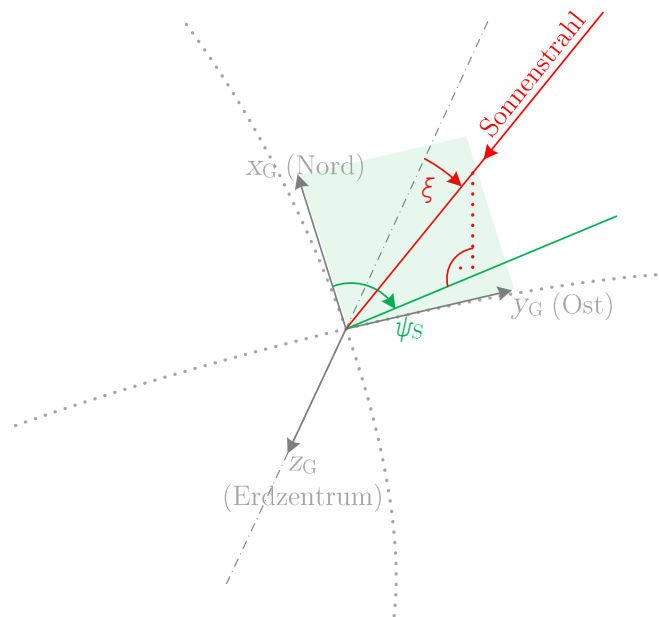


Abbildung 4.4: Azimut- und Zenitwinkel im geodätischen Koordinatensystem

Ein weiterer Parameter ist die Deklination der Sonne δ . Er beschreibt die Neigung der Erdachse im Jahresumlauf und schwankt um $\pm 23,5^\circ$. In der Jahreszeit des Sommers ist er positiv und zum Winter negativ. Der Zeitpunkt, wenn die Deklination null beträgt, wird als Frühlingspunkt bezeichnet und kennzeichnet den Frühlingsbeginn. Abbildung 4.5 zeigt schematisch den Verlauf der Deklination. Sie kann mit Formel 4.6 berechnet werden.

$$\delta = -23.5^\circ \cdot \cos\left(\frac{2 \cdot \pi}{365} \cdot (n + 10)\right) \tag{4.6}$$

mit: δ : Deklination der Sonne [rad]
 n : Tag im Jahr [-]

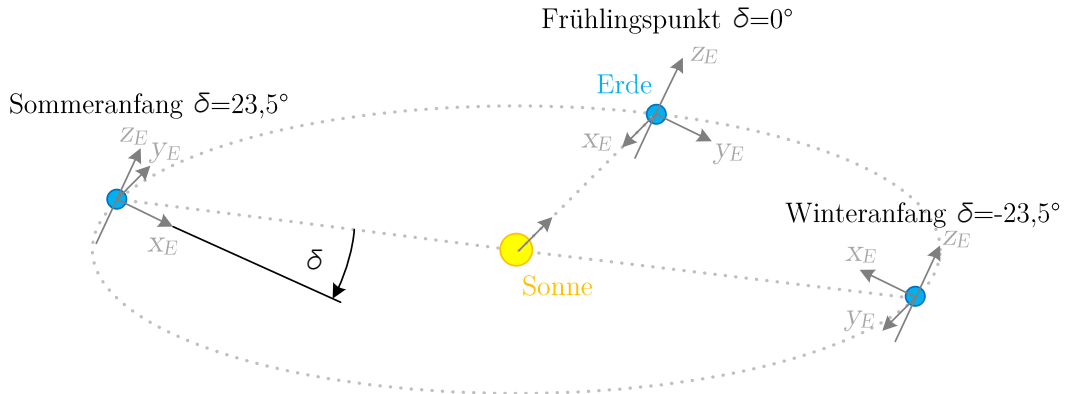


Abbildung 4.5: Deklination der Sonne im Jahresverlauf

Der Einfluss der Drehung um die Erdachse abhängig von der Zeit wird durch den Stundenwinkel τ dargestellt. Dieser berechnet sich nach Formel 4.7. Die Zeitangabe darin bezieht sich auf die Lokalzeit der entsprechenden Zeitzone im 24 *h*-System.

$$\tau = \frac{t}{12} \cdot \pi \quad (4.7)$$

mit: τ : Stundenwinkel der Sonne [rad]
 t : lokale Uhrzeit [h]

Die Abhängigkeit der Position wird durch den Längen- und Breitengrad des entsprechenden Missionsortes dargestellt. Der Längengrad λ , teilweise mit geographischer Länge bezeichnet, gibt die Position in Ost- oder Westrichtung zum Nullmeridian an. Der Breitengrad ϕ ist die nördliche oder südliche Distanz der Position ausgehend vom Äquator. Es handelt sich dabei um Winkelangaben.

Die Parameter des Stundenwinkels sowie der Längen- und Breitengrad sind zur Visualisierung in Abbildung 4.6 im erdfesten Koordinatensystem dargestellt.

Festlegung der Drehmatrizen

Die Richtung von Zenit- und Azimutwinkel lässt sich durch eine Kombination von elementaren Drehmatrizen bestimmen. Da diese in der Literatur nicht sehr präzise nachvollziehbar dargestellt wird, soll im Folgenden eine ausführliche Herleitung erfolgen. Ziel ist die Beschreibung des Sonnenstrahls im geodätischen Koordinatensystem. Bezüglich des Sonnenstrahls ist nur eine Richtung relevant, da nur die direkte Strahlung von Relevanz ist. Aufgrund des Verhältnis des Abstandes Sonne-Erde zu Erdradius kann der Winkeleinfluss durch den Erdradius vernachlässigt werden.

Die erste Drehoperation besteht aus einer Drehung um die Z-Achse des erdfesten Koordinatensystems (KOS), da der Sonnenstrahl entgegen der erdfesten x_E -Achse auftritt. Zur Verdeutlichung kann bereits gezeigte Abbildung 4.5 verwendet werden. Die Drehmatrix T_{1z} ist in Formel 4.8 aufgeführt. Die zweite Drehung T_{2y} erfolgt um die Y-Achse um den Einfluss der Deklination δ zu berücksichtigen und ist in Formel 4.9 gezeigt. Die dritte Drehoperation beschreibt die Drehung um die Z-Achse. Hier sind zwei Drehmatrizen bereits kombiniert, da die verwendeten Parameter um die gleiche Z-Achse drehen. Die Drehmatrix T_{3z} mit dem Längengrad λ und dem Stundenwinkel τ ist in 4.10 aufgeführt. Die letzte

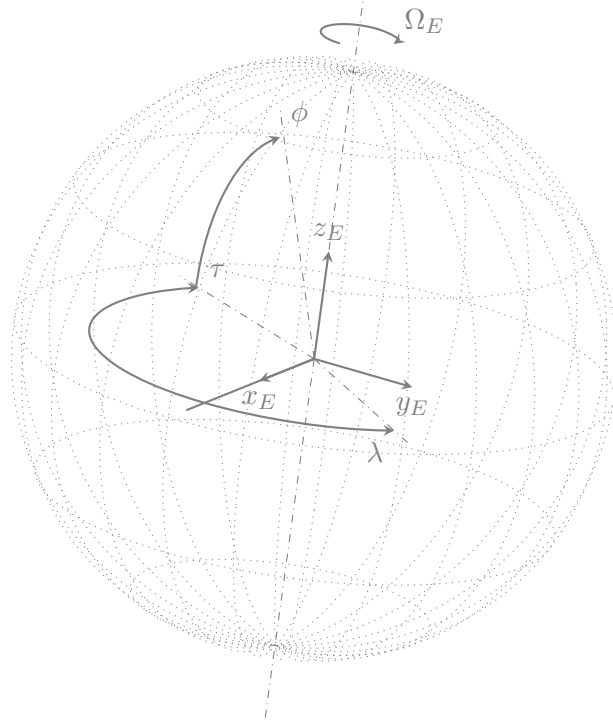


Abbildung 4.6: Drehwinkel vom erdfesten zum geodätischen KOS

Drehung wird um die Y-Achse ausgeführt. Durch die Definition des Breitengrades ϕ ist der Faktor $\pi/2$ hinzuzufügen [37]. Die Drehmatrix T_{4y} ist in 4.11 dargestellt.

$$T_{1z} = \begin{pmatrix} \cos \pi & \sin \pi & 0 \\ -\sin \pi & \cos \pi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.8)$$

$$T_{2y} = \begin{pmatrix} \cos \delta & 0 & -\sin \delta \\ 0 & 1 & 0 \\ \sin \delta & 0 & \cos \delta \end{pmatrix} \quad (4.9)$$

$$T_{3z} = \begin{pmatrix} \cos(\lambda + \tau) & \sin(\lambda + \tau) & 0 \\ -\sin(\lambda + \tau) & \cos(\lambda + \tau) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.10)$$

$$T_{4y} = \begin{pmatrix} \cos(-\frac{\pi}{2} - \phi) & 0 & -\sin(-\frac{\pi}{2} - \phi) \\ 0 & 1 & 0 \\ \sin(-\frac{\pi}{2} - \phi) & 0 & \cos(-\frac{\pi}{2} - \phi) \end{pmatrix} \quad (4.11)$$

Die Drehmatrizen können zusammen mit dem Sonnenvektor nach Formel 4.12 verrechnet werden.

$$T_{ge} = T_{4y} \cdot T_{3z} \cdot T_{2y} \cdot T_{1z} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (4.12)$$

Die daraus resultierende Transformationsmatrix T_{ge} gibt die Komponenten des Sonnenstrahls im geodätischen Koordinatensystem wieder. Sie ist in Formel 4.13 gezeigt.

$$T_{ge} = \begin{pmatrix} \sin \delta \cdot \sin(-\frac{\pi}{2} - \phi) - \cos \delta \cdot \cos(\lambda + \tau) \cdot \cos(-\frac{\pi}{2} - \phi) \\ \cos \delta \cdot \sin(\lambda + \tau) \\ -\cos(-\frac{\pi}{2} - \phi) \cdot \sin \delta - \cos \delta \cdot \cos(\lambda + \tau) \cdot \sin(-\frac{\pi}{2} - \phi) \end{pmatrix} = \begin{pmatrix} T_{ge}(1) \\ T_{ge}(2) \\ T_{ge}(3) \end{pmatrix} \quad (4.13)$$

Der Zenitwinkel ξ lässt sich durch die Einzelkomponenten von T_{ge} berechnen nach Formel 4.14. Der Höhenwinkel α_S bildet den Gegenwinkel zum Zenitwinkel.

$$\xi = \frac{\pi}{2} - \alpha_S = \frac{\pi}{2} - \arctan \left(\frac{T_{ge}(3)}{\sqrt{T_{ge}(1)^2 + T_{ge}(2)^2}} \right) \quad (4.14)$$

Der Azimut Ψ_S lässt sich ähnlich nach Formel 4.15 bestimmen. Aufgrund der Verwendung des arctan ist eine Quadrantenunterscheidung der Komponenten von T_{ge} durchzuführen. Dadurch muss der Azimut nachträglich nach Abbildung 4.7 angepasst werden, um diesen korrekt abzubilden. Der Azimut ist definiert mit $+180^\circ$ ausgehend von der x_g -Achse. Folglich muss für die Richtungen Südwest und -ost korrigiert werden.

$$\Psi_S = \arctan \left(\frac{T_{ge}(2)}{T_{ge}(1)} \right) \quad (4.15)$$

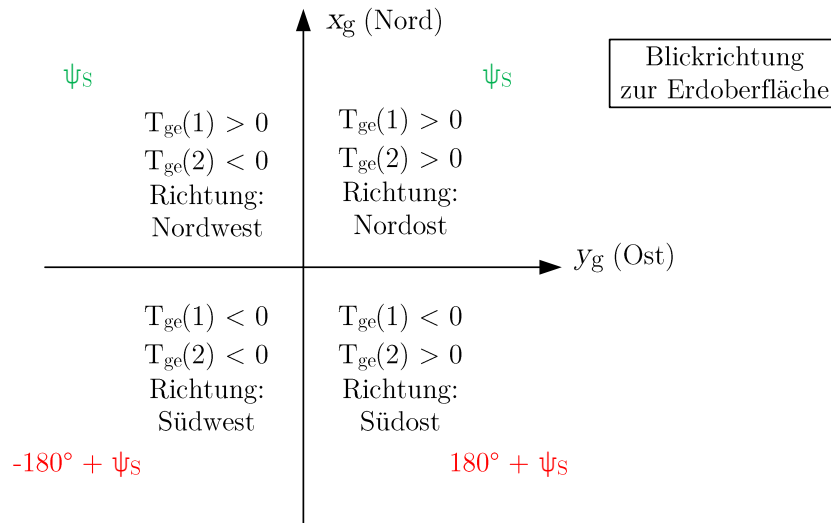


Abbildung 4.7: Azimutkorrektur abhängig vom Quadranten

4.1.3 Windmodell

Zum Aufbau eines Windmodells werden vorhandene Datensätze von [1] verwendet. Diese bestehen aus Windgeschwindigkeiten und der jeweiligen -richtung verteilt über die Flughöhe. Aufgrund der geplanten Missionsorte werden Daten zu den Standorten Hamburg, Kiruna und Yuma gewählt. Zusätzlich enthalten diese Datensätze Winddaten zu verschiedenen Monaten im Jahr. Abbildung 4.8 zeigt die Windgeschwindigkeiten der Stadt Hamburg in Deutschland. Neben den Windgeschwindigkeiten ist die projektierte Flugenveloppe in True Airspeed (TAS) angegeben. Außerdem ist der Flugbereich markiert, der den größten Teil der Mission abbilden soll. Dieser entspricht der über Wochen andauernden

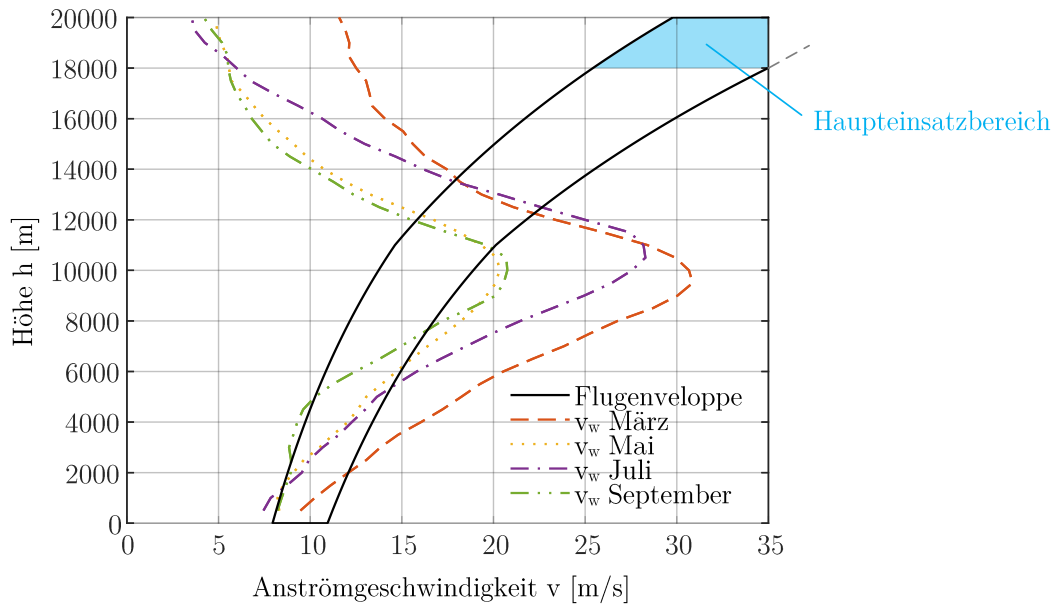


Abbildung 4.8: Windgeschwindigkeiten über Hamburg [1]

Platzierung in der Stratosphäre. Da hier die Windgeschwindigkeiten dauerhaft kleiner sind als die Fluggeschwindigkeit ist die Einhaltung einer vorgegebenen Flugbahn möglich. Jedoch ist für den Steigflug eine Abweichung der Flugbahn zu erwarten und nicht vermeidbar wie aus Abbildung 4.8 hervorgeht. Dieser Umstand gilt gleichermaßen für den Standort Kiruna in Schweden, dargestellt in Abbildung 4.9.

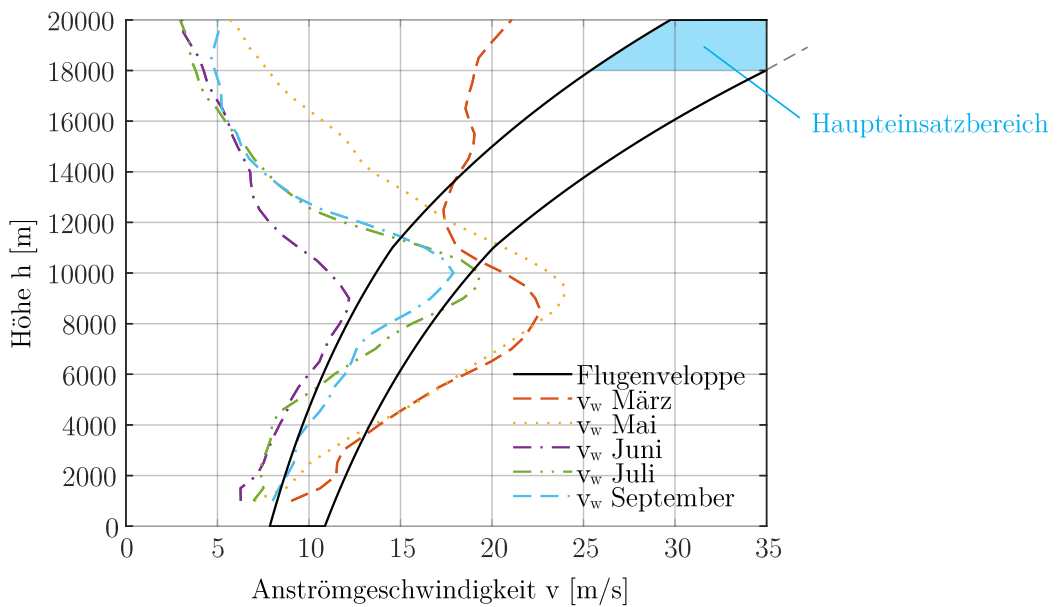


Abbildung 4.9: Windgeschwindigkeiten über Kiruna [1]

Lediglich bei der Durchführung der Mission in Yuma, eine Stadt im Bundesstaat Arizona der Vereinigten Staaten von Amerika, ist eine geforderte Flugbahn ohne größere Abweichungen abzufliegen.

In Abbildung 4.10 sind die Windgeschwindigkeiten stets kleiner als die Fluggeschwindigkeit. Für die anderen Standorten kann als Behelfslösung die Abweichung der Flugbahn durch entsprechende Vorkorrekturen in der geplanten Trajektorie teilweise kompensiert werden.

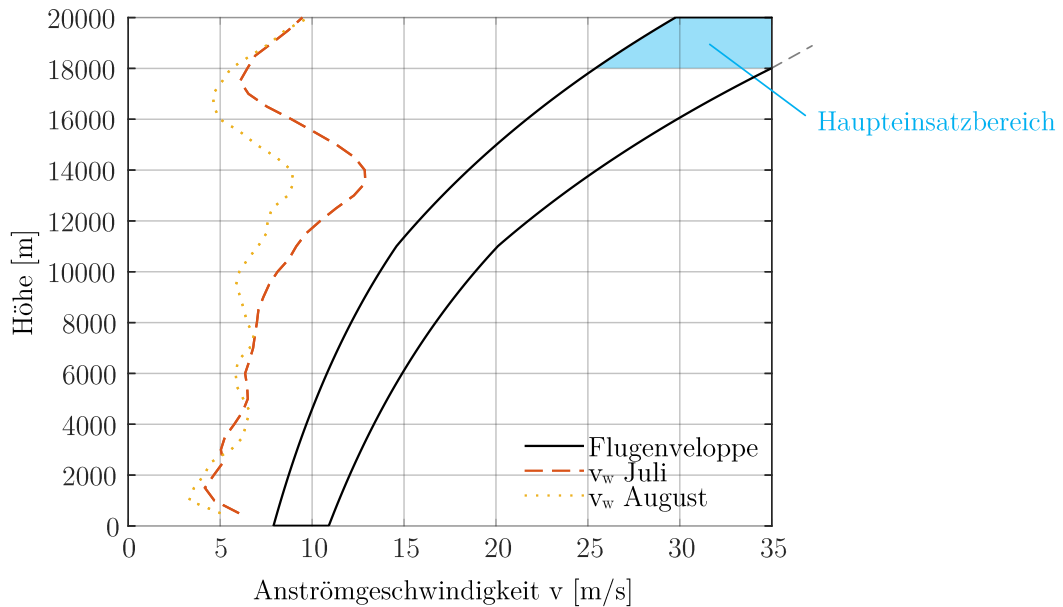


Abbildung 4.10: Windgeschwindigkeiten über Yuma [1]

Die Windrichtung wird für sämtliche Positionen konstant Richtung Nord gesetzt um Verzerrungen in der Flugbahn vorerst gleich zu belassen.

4.2 Flugzeug

Die Modellierung des Flugzeuges unterteilt sich durch den Umfang der abzubildenden Aufgaben in mehrere Abschnitte. Zuerst werden Koordinatensysteme, Fluggeschwindigkeiten und -abschnitte beschrieben. Diese Punkte gehören zu den Flugleistungen. Darauf folgt die Aerodynamik des Flügel und des Leitwerkes. Dazu gehören auch die Trimmrechnung und Widerstände am Flugzeug. Der Antriebsstrang bildet eine eigene Kategorie. Zuletzt wird der Solargenerator und die Energieverwaltung modelliert. Letzteres beinhaltet das Batteriemodell.

4.2.1 Koordinatensysteme

Zur Beschreibung des Flugzeuges im Raum werden verschiedene Koordinatensysteme (KOS) verwendet. Betreffend des Flugzeuges werden das geodätische und flugzeugfeste KOS verwendet entsprechend indiziert mit g und f . Die Achsen des geodätischen KOS sind wie folgt nach [37] definiert:

- x_g : senkrecht zur z -Achse, Richtung Norden
- y_g : senkrecht zur z -Achse, Richtung Osten
- z_g : Richtung Erdmittelpunkt, folglich senkrecht zur Erdoberfläche

Die Definition des flugzeugfesten Systems nach [37] ergibt sich zu:

- x_f : Richtung entlang der Flugzeuglängsachse, Ausrichtung zur Flugzeugnase
- y_f : senkrecht zur Symmetrie des Flugzeuges, Ausrichtung nach rechts
- z_f : senkrecht zur x-y-Ebene nach unten ausgerichtet

Beide KOS haben ihren Ursprung im Massenschwerpunkt des Flugzeuges. Die Transformation T_{fg} , also vom geodätischen zum flugzeugfesten KOS, kann nach Formel 4.16 berechnet werden. Die enthaltenen Drehmatrizen sind in Formel 4.17 4.18 4.19 gegeben.

$$T_{fg} = T_{x\phi} \cdot T_{y\theta} \cdot T_{z\Psi} \quad (4.16)$$

$$T_{x\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \quad (4.17)$$

$$T_{y\theta} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (4.18)$$

$$T_{z\Psi} = \begin{pmatrix} \cos \Psi & \sin \Psi & 0 \\ -\sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.19)$$

mit: T_{fg} : Transformationsmatrix vom geodätischen in flugzeugfestes KOS
 $T_{x\phi}$: Drehmatrix um X-Achse mit dem Winkel ϕ
 $T_{y\theta}$: Drehmatrix um Y-Achse mit dem Winkel θ
 $T_{z\Psi}$: Drehmatrix um Z-Achse mit dem Winkel Ψ
 ϕ : Rollwinkel [rad]
 θ : Nickwinkel [rad]
 Ψ : Gierwinkel [rad]

Zur Vermeidung von Verwechslungen mit dem Breitengrad ϕ wird der Rollwinkel folgend mit ϕ_{roll} bezeichnet. Der Nickwinkel θ setzt sich zusammen aus dem Anstellwinkel des Flugzeuges und dem Bahnneigungswinkel, gezeigt in Formel 4.20 aus [37].

$$\theta = \gamma + \alpha \quad (4.20)$$

mit: γ : Bahnneigungswinkel [rad]
 α : Anstellwinkel [rad]

In Abbildungen der Folgekapitel werden teilweise Achsen des aerodynamischen KOS gezeigt. Diese sind mit a indiziert. Es unterscheidet sich zum flugzeugfesten KOS durch den Anstellwinkel α und den Schiebewinkel β . Da diese jedoch sehr klein sind, wird in Berechnungen der Einfluss dieser vernachlässigt bzw. nur in der Ebene berücksichtigt.

4.2.2 Fluggeschwindigkeit

Die Fluggeschwindigkeit wird in Equivalent Airspeed (EAS) angegeben und steht für die äquivalente Fluggeschwindigkeit. Sie wird als Vergleichsgröße verwendet, da bei gleicher EAS in unterschiedlichen Höhen der gleiche Auftrieb entsteht [38] [39]. Für Berechnungen, beispielsweise der exakten Position, ist jedoch die True Airspeed (TAS) zu verwenden. Diese entspricht der wahren Geschwindigkeit am Flugzeug. Sie kann mit Formel 4.21 nach [40] aus der EAS berechnet werden.

$$v_{TAS} = v_{EAS} \cdot \sqrt{\frac{\rho_0}{\rho(h)}} \quad (4.21)$$

mit: v_{TAS} : Fluggeschwindigkeit in TAS [m/s]
 v_{EAS} : Fluggeschwindigkeit in EAS [m/s]
 $\rho(h)$: Luftdichte [kg/m^3]
 $\rho_0 = 1,225 \text{ kg}/m^3$: Luftdichte auf Meereshöhe

4.2.3 Flugabschnitte

Zur Durchführung der Mission sind verschiedene Flugmanöver erforderlich, die abhängig von den Bahnparametern eine Einteilung des Fluges in Abschnitte bzw. Segmente erlauben. Diese Abschnitte lassen sich in Steig- und Sinkflug, sowie dem Horizontalflug einteilen. Des Weiteren ist die Unterscheidung zwischen Geradeaus- und Kurvenflug erforderlich. Ziel ist die Beschreibung des Auftriebskoeffizienten c_L in Abhängigkeit zu den Bahnparametern.

Nach Abbildung 4.11 ergibt sich das Kräftegleichgewicht im aerodynamischen KOS mit Formel 4.22 und 4.23. Dabei liegen die Annahmen nach [43] und [40] entsprechend Tabelle 4.2 zugrunde. Die Gewichtskraft lässt sich nach Formel 4.24 in ihren Komponente angeben.

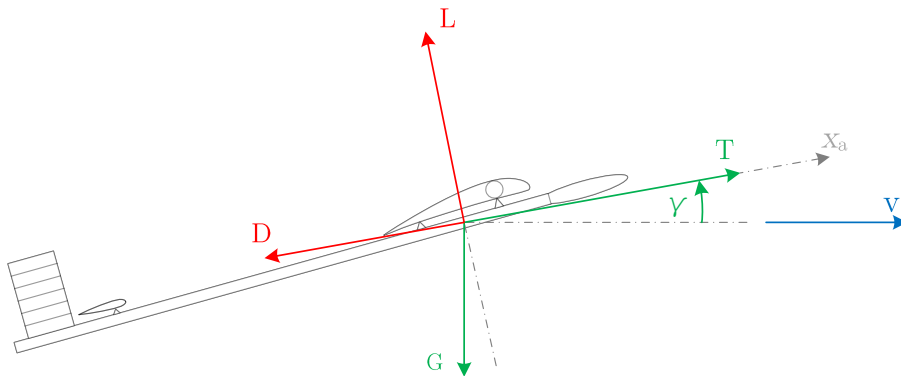


Abbildung 4.11: Kräfte am Flugzeug im Flug

$$T - D - m \cdot g \cdot \sin \gamma = 0 \quad (4.22)$$

$$L - m \cdot g \cdot \cos \gamma = 0 \quad (4.23)$$

$$G = m \cdot g \quad (4.24)$$

mit:	D	: Widerstandskraft [N]
	L	: Auftriebskraft [N]
	T	: Schubkraft [N]
	m	: Flugzeugmasse [kg]
	g	: Erdbeschleunigung [m/s^2]
	γ	: Bahnneigungswinkel [rad]
	G	: Gewichtskraft [N]

Annahme	mit:
$T > D$	
\dot{v}	: Beschleunigung [m/s^3]
$\dot{\gamma}$: Nickrate [rad/s]
$\cos \alpha \approx 1$	
$\sin \alpha \approx 0$	

Tabelle 4.2: Annahmen bezüglich Kräftegleichgewicht

Der Auftrieb L kann mit Formel 4.25 nach [43] berechnet werden und in Formel 4.23 eingesetzt werden. Dadurch ergibt sich der Auftriebskoeffizient nach Formel 4.26.

$$L = \frac{\rho}{2} \cdot v^2 \cdot c_L \cdot S \quad (4.25)$$

$$c_L = \frac{2 \cdot m \cdot g}{\rho \cdot v^2 \cdot S} \cdot \cos \gamma \quad (4.26)$$

mit:	ρ	: Luftdichte [kg/m^3]
	c_L	: Auftriebskoeffizient [-]
	S	: Flügelfläche [m^2]

Die Widerstandskraft D lässt sich mit Formel 4.27 nach [43] berechnen und kann in 4.22 eingesetzt werden, wodurch sich der Schub T in Formel 4.28 ergibt.

$$D = \frac{\rho}{2} \cdot v^2 \cdot c_D \cdot S \quad (4.27)$$

$$T = \frac{\rho}{2} \cdot v^2 \cdot c_D \cdot S + m \cdot g \cdot \sin \gamma \quad (4.28)$$

mit:	ρ	: Luftdichte [kg/m^3]
	c_D	: Widerstandskoeffizient [-]
	S	: Flügelfläche [m^2]

Alternativ kann laut [43] der Schub durch Umformung mit Formel 4.29 berechnet werden.

$$T = \left(\sin \gamma + \cos \gamma \cdot \frac{c_D}{c_L} \right) \cdot m \cdot g \quad (4.29)$$

Die bisher dargelegten Formeln bilden den geraden Flug ab. Diese können verwendet werden für die Segmente Steig- und Sinkflug sowie für den Horizontalflug. Für den Kurvenflug sind weitere Parameter zu integrieren. Dabei erhöht sich der notwendige Auftrieb umgekehrt proportional zum \cos des Rollwinkels. Dies wird durch den Lastfaktor n ausgedrückt, dargestellt in Formel 4.30 nach [43].

$$n = \frac{1}{\cos(\phi_{roll})} \quad (4.30)$$

Der Lastfaktor wird in die Berechnung des Auftriebes integriert nach Formel 4.31.

$$L = n \cdot G = \frac{1}{\cos(\phi_{roll})} \cdot G \quad (4.31)$$

Dies kann in Formel 4.26 zur Berechnung des Auftriebskoeffizienten eingesetzt werden. Dadurch ergibt sich Formel 4.32 zur Berechnung des Auftriebskoeffizienten abhängig vom Bahnneigungs- und Rollwinkel.

$$c_L = \frac{2 \cdot m \cdot g}{\rho \cdot v^2 \cdot S} \cdot \frac{\cos \gamma}{\cos \phi_{roll}} \quad (4.32)$$

mit:	ρ	: Luftdichte [kg/m^3]
	c_L	: Auftriebskoeffizient [-]
	S	: Flügelfläche [m^2]
	m	: Flugzeugmasse [kg]
	g	: Erdbeschleunigung [m/s^2]
	v	: Fluggeschwindigkeit [m/s]
	γ	: Bahnneigungswinkel [rad]
	ϕ_{roll}	: Rollwinkel [rad]

Der Rollwinkel ϕ_{roll} wird bestimmt aus der Vorgabe des Radius des zu fliegenden Flugkreises und der Fluggeschwindigkeit. Er kann aus Formel 4.33 nach [40] berechnet werden mit Formel 4.34.

$$r_{turn} = \frac{v^2}{g \cdot \tan(\phi_{roll})} \quad (4.33)$$

$$\phi_{roll} = \arctan\left(\frac{v^2}{g \cdot r_{turn}}\right) \quad (4.34)$$

mit:	r_{turn}	: Kurvenradius des geflogenen Kreises [m]
------	------------	---

Der Einfluss des Kurvenradius auf den Schub ergibt sich automatisch durch die Erhöhung des Auftriebskoeffizienten in Formel 4.29.

4.2.4 Flügel und Leitwerke

Flügel

Die aerodynamischen Parameter des Hauptflügels werden durch Kennfelder von [1] beschrieben. Der Hauptflügel des Flugzeuges HAP verwendet das Flügelprofil MH139R von [1]. Abbildung 4.12 zeigt die entsprechende Profilpolare. Die Werte sind für das Flügelprofil. Es ist anzumerken, dass aufgrund des Simulationsablaufs zuerst der Auftriebskoeffizient c_L nach Formel 4.32 berechnet wird. Der Auftriebskoeffizient des Flügelprofils c_l wird aus dem Auftriebskoeffizient des gesamten Flügels c_L berechnet. Für die Anwendung im Programm sind beide Richtungen der Berechnung (von c_L zu c_l und umgekehrt) notwendig. Nachfolgende Formeln können dazu verwendet und jeweils umgestellt werden. Zur Berechnung des Profilauftriebskoeffizienten c_l kann Formel 4.35 aus [41] verwendet werden. Diese gilt

für Flügel mit hoher Streckung, großem Nasenradius und geringer Pfeilung. Eine genauere Bestimmung für ungepfeilte Rechteckflügel erlaubt Formel 4.36 aus [42]. Die darin enthaltene Streckung Λ wird mit Formel 4.37 aus [40] berechnet. Für den Flügel des Flugzeuges HAP ergibt sich ein Aufschlag von ca. 8,7 % auf den Auftriebskoeffizienten des Flügelprofils.

$$c_L = 0,9 \cdot c_l \cdot \cos \varphi_{l/4} \quad (4.35)$$

$$c_L = c_l \cdot \frac{\Lambda}{\lambda + 2} \quad (4.36)$$

$$\Lambda = \frac{b_i^2}{S_i} \quad (4.37)$$

mit: c_L : Auftriebskoeffizient des Flügels [-]
 c_l : Auftriebskoeffizient des Flügelprofils [-]
 $\varphi_{l/4} = 0$: Pfeilung des Flügels [rad]
 Λ : Streckung [-]
 b_i : Spannweite des jeweiligen Flügels [m]
 S_i : Flügelfläche des jeweiligen Flügels [m²]

Ausgehend von einem Auftriebskoeffizienten c_l für das Flügelprofil kann der Nullwiderstandsbeiwert c_{D0} durch Interpolation der Kurvenschar bestimmt werden wie in Abbildung 4.12 angedeutet. Dazu muss die vorliegende Reynoldszahl berechnet werden. Diese wird nach Formel 4.38 aus [43] berechnet.

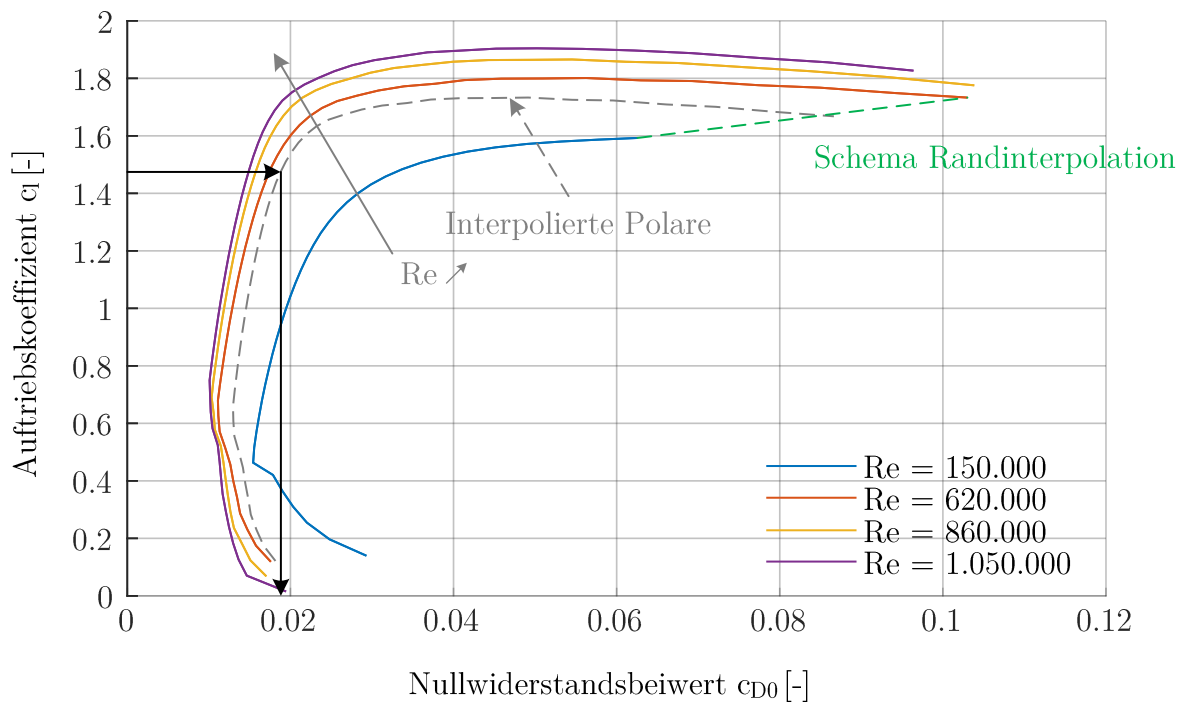


Abbildung 4.12: Profilpolare des Flügels für MH139R nach [1]

$$Re = \frac{v \cdot c}{\nu(h)} \quad (4.38)$$

mit: Re : Reynoldszahl [-]
 v : Fluggeschwindigkeit [m/s]
 c : Referenzlänge des Flügel [m]
 $\nu(h)$: kinematische Viskosität [m^2/s]

In Abbildung 4.12 ist zusätzlich verdeutlicht wie die Ränder einer interpolierten Polaren erzeugt werden. Die Unter- oder Überschreitung der Polaren mit kleinster und größter Reynoldszahl ist nicht möglich. Aufgrund der Platzierung der Solarzellen auf der Flügeloberseite wird laut [1] ein Proportionalitätsfaktor mit dem Nullwiderstand verrechnet. Im aktuellen Fall beträgt dieser 25 %.

Zur Berechnung des Widerstandskoeffizienten des Flügels c_D wird eine asymmetrische Polare angenommen nach Formel 4.39 aus [43]. Der Oswaldfaktor e ist gegeben durch [1]. Bei Verwendung einer symmetrischen Polare, gezeigt in Formel 4.40, kann dies für das vorliegende Flugzeug und den zu erwartenden Auftriebskoeffizienten eine Erhöhung des induzierten Widerstandsanteils um den Faktor 2,7-4 ergeben.

$$c_D = c_{D0} + c_{Di} = c_{D0} + \frac{(c_l - c_{l=0})^2}{\pi \cdot \Lambda \cdot e} \quad (4.39)$$

$$c_D = c_{D0} + c_{Di} = c_{D0} + \frac{c_l^2}{\pi \cdot \Lambda \cdot e} \quad (4.40)$$

mit: c_D : Widerstandskoeffizient [-]
 c_{D0} : Nullwiderstandsbeiwert [-]
 c_{Di} : induzierter Widerstand [-]
 c_l : Auftriebskoeffizient des Flügelprofils[-]
 $c_{l=0}$: Nullauftriebskoeffizient [-]
 Λ : Streckung [-]
 $e = 0,91$: Oswaldfaktor [-]

Der Anstellwinkel α wird gleichermaßen ermittelt wie zuvor. Im Kennfeld von Abbildung 4.13 wird mit dem Auftriebsbeiwert abhängig von der Reynoldszahl der Anstellwinkel ermittelt. Umgekehrt kann für einen Anstellwinkel von 0° der für die Polare benötigte Nullauftriebskoeffizient $c_{l=0}$ ermittelt werden. Weiterhin kann durch den zuvor ermittelten Anstellwinkel in Abbildung 4.14 der entsprechende Wert für den Momentenbeiwert des Flügelprofils c_m ermittelt werden.

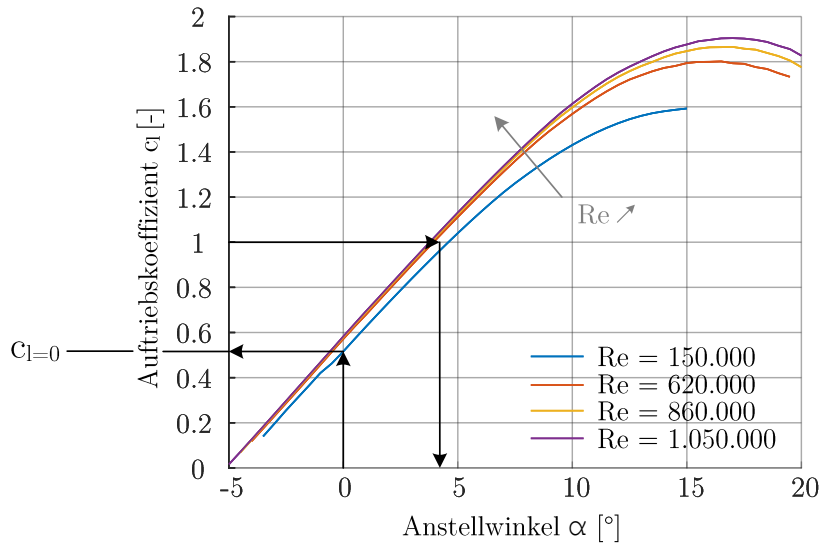


Abbildung 4.13: Auftriebskoeffizient über Anstellwinkel für MH139R nach [1]

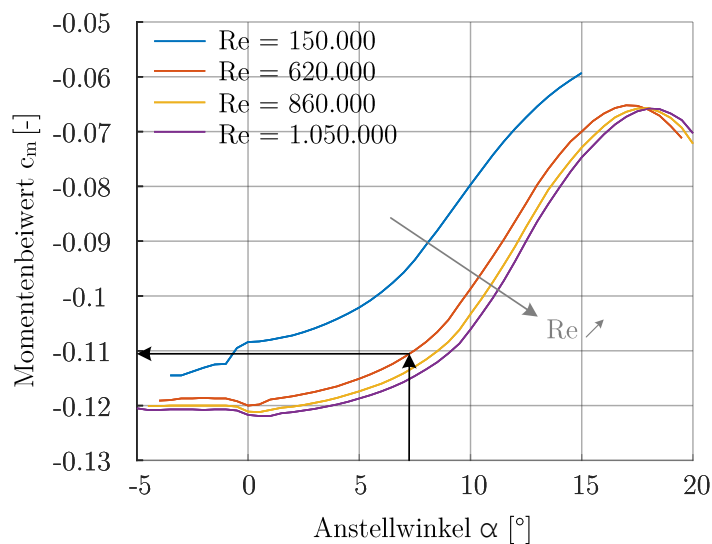


Abbildung 4.14: Momentenbeiwert über Anstellwinkel für MH139R nach [1]

Der Momentenbeiwert des Flügelprofils c_m kann mit Formel 4.41 aus [44] für den gesamten Flügel berechnet werden. Im vorliegenden Fall vereinfacht sich die Formel zu 4.42. Dadurch entsteht ein Proportionalitätsfaktor von ca. 0,91 für den Momentenbeiwert des Flügels c_M von HAP.

$$c_M = \frac{\Lambda \cdot (\cos \varphi_{l/4})^2}{\Lambda + 2 \cdot \cos \varphi_{l/4}} \cdot 0,5 \cdot (c_{m0,i} + c_{m0,a}) + \frac{\Delta c_{m0}}{\varepsilon_t} \cdot \varepsilon_t \quad (4.41)$$

$$c_M = \frac{\Lambda}{\Lambda + 2} \cdot c_m \quad (4.42)$$

mit:	c_M	: Momentenbeiwert des Flügels [-]
	c_m	: Momentenbeiwert des Flügelprofils [-]
	$c_{m0,i}$: Nullmomentenbeiwert des Flügelprofils außen [-]
	$c_{m0,a}$: Nullmomentenbeiwert des Flügelprofils außen [-]
	$\frac{\Delta c_{m0}}{\varepsilon_t} = 0$: Auswirkung einer linearen Verwindung [-]
	$\varepsilon_t = 0$: Verwindung des Flügels [rad]

Im betrachteten Fall werden sehr große Kurven geflogen, wodurch der Rollwinkel nur wenige Grad annimmt. Daher wird der Einfluss auf Widerstands- und Auftriebsbeiwert durch eventuelle Querrüderausschläge vernachlässigt.

Leitwerk

Das verwendete Leitwerk ist als Pendelleitwerk ausgelegt. Folglich verändert das gesamte Leitwerk seinen Anstellwinkel in Abhängigkeit des benötigten Auftriebes. Das Verfahren zur Ermittlung der aerodynamischen Größen bleibt identisch wie beim Flügel. Jedoch wird eine symmetrische Polare verwendet nach Formel 4.40. Die verwendeten Kennfelder zum Flügelprofil NACA0009 des Pendelleitwerks sind in Anhang A zu finden.

Der Wert des Oswaldfaktors e unterscheidet sich zum Flügel, da dieser für das Leitwerk analytisch ermittelt wird. Der Oswaldfaktor e wird mit 0,93 bestimmt aus [43] für einen Rechteckflügel.

4.2.5 Trimmrechnung

Die Trimmung bezeichnet die dauerhafte Anstellung des Höhenleitwerks zur Herstellung eines momentenfreien Zustandes um den Schwerpunkt des Flugzeuges. Der Schwerpunkt oder auch Center of Gravity (CoG) dient als Drehpunkt für die Aufstellung des Momentengleichgewichts. Dies wird in Abbildung 4.15 dargestellt. Formel 4.43 beschreibt den getrimmten Zustand bei dem die Summe aller Momente zu null wird.

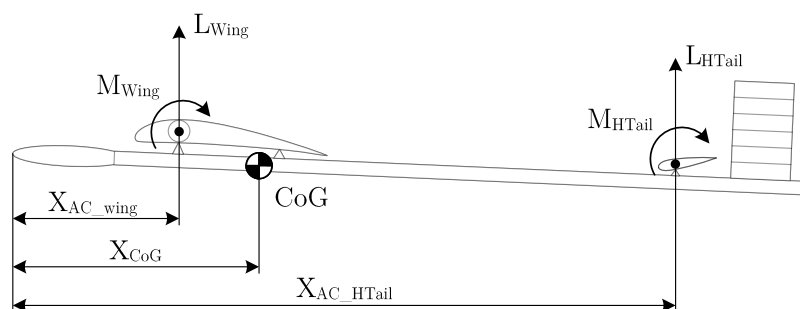


Abbildung 4.15: Momentengleichgewicht um den Schwerpunkt CoG

$$L_{Wing} \cdot (x_{CoG} - x_{ACWing}) + M_{Wing} - L_{HTail} \cdot (x_{ACWing} - x_{CoG}) + M_{HTail} = 0 \quad (4.43)$$

mit:	x_{CoG}	: Abstand des Schwerpunkts zur Nase [m]
	x_{ACWing}	: Abstand des Neutralpunktes des Flügels zur Nase [m]
	$x_{ACHTail}$: Abstand des Neutralpunktes des Höhenleitwerks zur Nase [m]
	L_{Wing}	: Auftrieb des Flügels [N]
	M_{Wing}	: Moment des Flügels [Nm]
	L_{HTail}	: Auftrieb des Höhenleitwerks [N]
	M_{HTail}	: Moment des Höhenleitwerks [Nm]

Das Moment des Höhenleitwerks M_{HTail} ist verschwindend gering, weshalb es zu null gesetzt wird. Das Gleichgewicht kann nach dem Auftrieb am Höhenleitwerk umgestellt werden, dargestellt in Formel 4.44 . Die Berechnung einzelner Komponenten ist in Formel 4.45 4.46 und 4.47 gegeben.

$$L_{HTail} = \frac{L_{Wing} \cdot (x_{CoG} - x_{ACWing}) + M_{Wing}}{(x_{ACWing} - x_{CoG})} \quad (4.44)$$

$$L_{Wing} = q \cdot c_L \cdot S_{Wing} \quad (4.45)$$

$$M_{Wing} = q \cdot c_M \cdot S_{Wing} \cdot c_{Wing} \quad (4.46)$$

$$q = \frac{\rho}{2} \cdot v^2 \quad (4.47)$$

mit:	q	: Staudruck [N/m^2]
	c_L	: Auftriebskoeffizient [-]
	c_M	: Momentenbeiwert [-]
	c_{Wing}	: Bezugsflügelteufe des Flügels [m]
	S_{Wing}	: Flügelfläche [m^2]

Somit ergibt sich nach Umstellen und Einsetzen Formel 4.48 für die Berechnung des Leitwerksauftriebes.

$$L_{HTail} = q \cdot S_{Wing} \cdot \frac{c_L \cdot (x_{CoG} - x_{ACWing}) + c_M \cdot c_{Wing}}{(x_{ACWing} - x_{CoG})} \quad (4.48)$$

Der Auftrieb am Leitwerk erhöht den benötigten Auftrieb am Flügel nach Formel 4.49 , Folglich ändern sich die Kräfte am Flügel und somit der Momentenhaushalt. Die Trimmrechnung muss daher iterativ durchgeführt werden bis eine geeignete Differenz erreicht ist.

$$L_{Wing} = L_{req} - L_{HTail} \quad (4.49)$$

mit:	L_{req}	: benötigter Auftrieb am Flugzeug [N]
------	-----------	---------------------------------------

4.2.6 Widerstände

Widerstände am Flugzeug werden grundsätzlich durch den Widerstandsbeiwert c_D ausgedrückt. Der Widerstandsbeiwert des gesamten Flugzeuges ergibt sich durch die Aufsummierung der Einzelwiderstände in Abhängigkeit des Verhältnis ihrer bespülten Oberfläche zur Flügelfläche [45]. Dieser Zusammenhang ist in Formel 4.50 nach [45] gezeigt.

$$c_D = c_{D_i} \cdot \frac{S_{wet i}}{S_{Wing}} + c_{D_{i+1}} \cdot \frac{S_{wet i+1}}{S_{Wing}} + \dots \quad (4.50)$$

mit: c_D : Widerstandsbeiwert [-]
 c_{D_i} : Widerstandsbeiwert der Komponente i [-]
 $S_{wet i}$: bespülte Oberfläche der Komponente i [m^2]
 S_{Wing} : Flügelfläche [m^2]

Die Widerstände des Flügels und des Höhenleitwerks können mit den Methoden der vorangegangenen Kapitel bestimmt werden. Weitere Widerstandserzeuger sind das Seitenleitwerk, der Rumpf, das Fahrwerk und die Antennen. Für diese sind Widerstandsbeiwerte von [1] gegeben.

Das Seitenleitwerk wird betrachtet als würde es aerodynamisch keinen Auftrieb erzeugen. Dazu wird ein schiebefreier Flug angenommen. Dieser ist energetisch auch sinnvoll, da so minimaler Widerstand durch das Seitenleitwerk und andere schiebende Komponenten erzeugt wird.

Der Widerstandsbeiwert des Rumpfes sollte nach [45] erweitert werden durch Form- und Interferenzfaktor. Formel 4.51 zeigt die entsprechende Berechnung nach [45].

$$c_{D fus} = \frac{S_{wet fus}}{S_{Wing}} \cdot c_{D0 fus} \cdot FF_{fus} \cdot IF_{fus} \quad (4.51)$$

mit: $c_{D fus}$: Widerstandsbeiwert des Rumpfes [-]
 $c_{D0 fus}$: Nullwiderstandsbeiwert des Rumpfes [-]
 FF_{fus} : Formfaktor Rumpf [-]
 IF_{fus} : Interferenzfaktor Rumpf [-]

Für den Formfaktor FF_{fus} wird Formel 4.52 angesetzt. Dadurch ergibt sich ein Wert von 1,06. Der Interferenzfaktor IF_{fus} wird auf 1,2 gewählt, da es sich um einen Hochdecker ohne Verkleidung handelt. Resultierend daraus erhöht sich der Widerstandsbeiwert des Rumpfes um ca. 27,2 %.

$$FF_{fus} = 1 + \frac{60}{(l_{fus}/d_{fus})^3} + \frac{l_{fus}/d_{fus}}{400} \quad (4.52)$$

mit: l_{fus} : Länge des Rumpfes [m]
 d_{fus} : Durchmesser des Rumpfes [m]

Für das Fahrwerk und die Antennen werden keine weiteren Modifikationen vorgenommen. Es werden die vorgegebenen Werte gewählt.

4.2.7 Antriebsstrang

Der Antriebsstrang wird beschrieben durch zwei weitere Komponenten bestehend aus dem Propeller und dem Motor. Der Propeller erzeugt den aerodynamischen Schub um den Vortrieb des Flugzeuges zu erzeugen. Der Motor erzeugt die notwendige Wellenleistung für die Abtriebswelle, auf der der Propeller befestigt ist. Die Modellierung beider Komponenten erfolgt separat und wird anschließend durch den gemeinsamen Betriebspunkt zusammengeführt.

Zur Bestimmung der notwendigen Antriebsleistung ist der Schub nach Formel 4.29 notwendig. Die entsprechende Formel ist hier mit dem erforderlichen Gesamtschub T_{req} in 4.53 angegeben.

$$T_{req} = (\sin \gamma + \cos \gamma \cdot \frac{c_D}{c_L}) \cdot m \cdot g \quad (4.53)$$

Der benötigte Schub eines Antriebes wird durch die Anzahl der Antriebsstränge aufgeteilt nach Formel 4.54.

$$T_i = \frac{T_{req}}{n_{eng}} \quad (4.54)$$

mit: T_{req} : erforderlicher Gesamtschub [N]
 T_i : Schub eines Antriebes [N]
 n_{eng} : Anzahl der Antriebe [-]

Propeller

Beim verwendeten Propeller handelt es sich um einen Propeller mit festem Steigwinkel. Die Steigung ist für den Betrieb im Reiseflug auf Reiseflughöhe ausgelegt. Die Leistungsparameter des Propellers sind durch vorhandene Datensätze zum Fortschrittgrad J , dem Schubbeiwert c_T und dem Leistungskoeffizienten c_P aus [1] festgelegt. Diese sind in Abbildung 4.16 aufgeführt. Der darin aufgeführte Propellerwirkungsgrad η_{Prop} kann mit Formel 4.55 nach [46] berechnet werden.

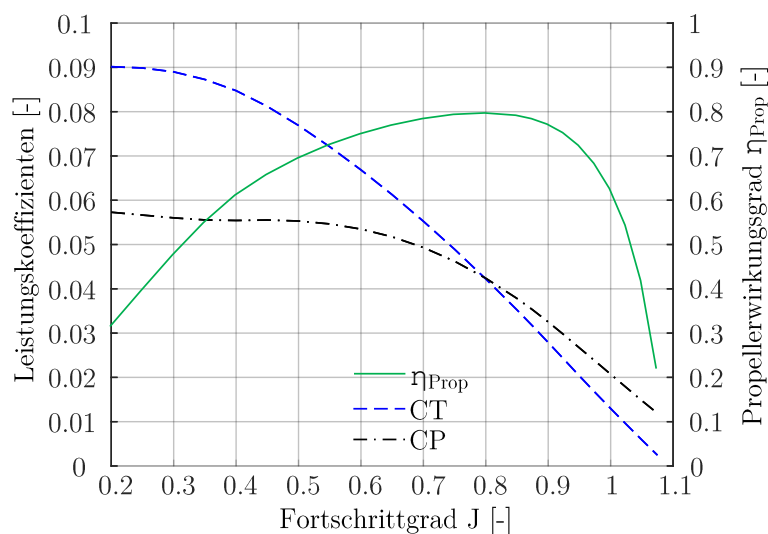


Abbildung 4.16: Propellerkennlinie über den Fortschrittgrad J nach [1]

$$\eta_{Prop} = J \cdot \frac{c_T}{c_P} \quad (4.55)$$

mit: η_{Prop} : Wirkungsgrad des Propeller [-]
 J : Fortschrittgrad [-]
 c_T : Schubbeiwert [-]
 c_P : Leistungskoeffizienten [-]

Der aerodynamische Schub des Propellers berechnet sich nach Formel 4.56 aus [46]. Die darin enthaltene Drehzahl des Propellers berechnet sich mit Formel 4.57. Mit dem Schub und der Fluggeschwindigkeit ist die aerodynamische Propellerleistung in Formel 4.58 nach [46] berechnet.

$$T_{Prop} = c_T \cdot \rho \cdot n_{Prop}^2 \cdot D_{Prop}^4 \quad (4.56)$$

$$n_{Prop} = \frac{v}{J \cdot D_{Prop}} \quad (4.57)$$

$$P_{Prop} = T_{Prop} \cdot v \quad (4.58)$$

mit: T_{Prop} : Schub des Propellers [N]
 ρ : Luftdichte [kg/m^3]
 n_{Prop} : Drehzahl des Propellers [$1/s$]
 D_{Prop} : Durchmesser des Propellers [m]
 v : Fluggeschwindigkeit [m/s]
 P_{Prop} : aerodynamische Leistung des Propellers [W]

Der Leistungskoeffizient koppelt die aerodynamische Leistung mit der Wellenleistung, wodurch diese mit Formel 4.59 nach [46] berechnet wird. Das Drehmoment an der Abtriebswelle kann folglich mit Formel 4.60 bestimmt werden.

$$P_{Prop shaft} = \rho \cdot n_{Prop}^3 \cdot D_{Prop}^5 \cdot c_P \quad (4.59)$$

$$M_{Prop} = \frac{P_{Prop shaft}}{2 \cdot \pi \cdot n_{Prop}} \quad (4.60)$$

mit: $P_{Prop shaft}$: Wellenleistung des Propellers [W]
 M_{Prop} : Drehmoment des Propellers [Nm]

Dadurch ist der Propeller komplett beschrieben. In Formel 4.61 ist nochmal verdeutlicht von welchen Faktoren der Propellerschub abhängig ist. Abbildung 4.17 zeigt den Propellerschub in Abhängigkeit vom Fortschrittgrad.

$$T_{Prop}(c_T, n_{Prop}(v, J, D_{Prop}), \rho(h), D_{Prop}) \quad (4.61)$$

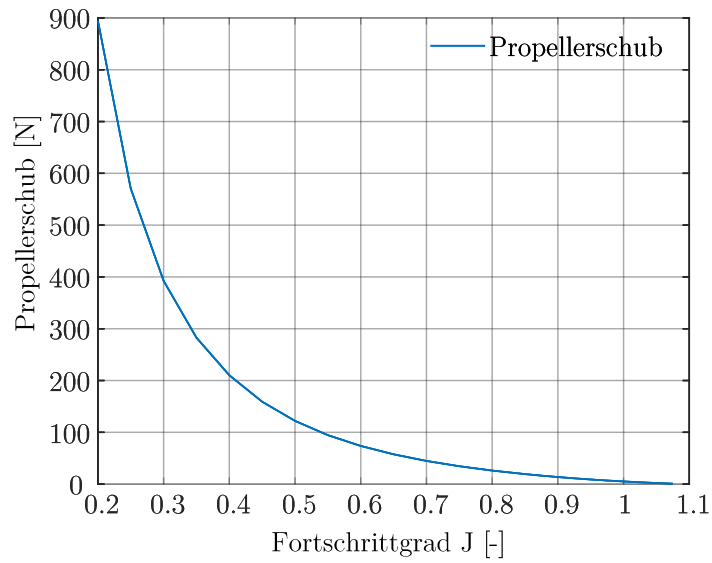


Abbildung 4.17: Schubverlauf über den Fortschrittgrad J

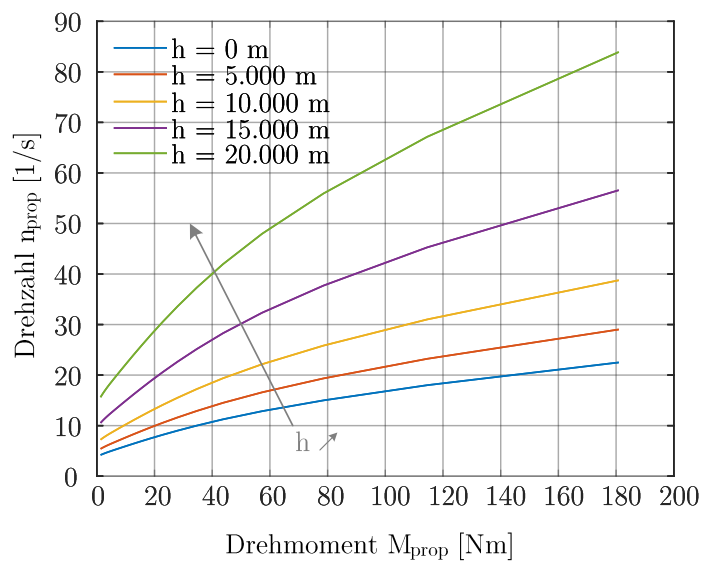


Abbildung 4.18: Drehzahl des Propellers über dem Drehmoment abhängig von der Flughöhe

Abbildung 4.18 zeigt das Drehmoment des Propellers über der Propellerdrehzahl. Dies wird im Folgenden als Propellerkennlinie bezeichnet und später wieder aufgegriffen.

(Anmerkung Formeln: Die dargelegten Formeln aus [46] sind in abweichender Indizierung in [43] zu finden)

Motor

Der Motor wird entsprechend eines Elektromotors modelliert. Das bedeutet, dass er abhängig von seinem Motorkennfeld in einem Betriebspunkt arbeitet. Die Beschreibung dieses Kennfeldes erfolgt durch die Parameter aufgelistet in Tabelle 4.3.

Parameter	Kürzel	Wert	Einheit	Bemerkung
max. Leistung	$P_{max\ Mot}$	2000	W	
max. Leistung bei Rekuperation	$P_{max\ rcu\ Mot}$	1000	W	
max. Drehmoment	$M_{max\ Mot}$	15	Nm	
min. Drehmoment	$M_{min\ Mot}$	-15	Nm	
Drehmomentkonstante	$k_M\ Mot$	0,2	$1/Nms$	
Leerlaufdrehzahl	$n_0\ Mot$	50	$1/s$	
elektr. Wirkungsgrad	$\eta_{el\ Mot}$	0,9	–	

Tabelle 4.3: Motorkenndaten des Flugzeuges HAP

Zur Erstellung des Kennfeldes ist der Motorbetrieb zu charakterisieren. Man unterscheidet den Last- und Generatorbetrieb abhängig davon ob der Motor Arbeit leistet oder Energie generiert [47]. Dieser Zusammenhang lässt sich in Abbildung 4.19 durch die Vorzeichen von Drehzahl und Drehmoment in Quadranten einteilen und entsprechend im Programm hinterlegen.

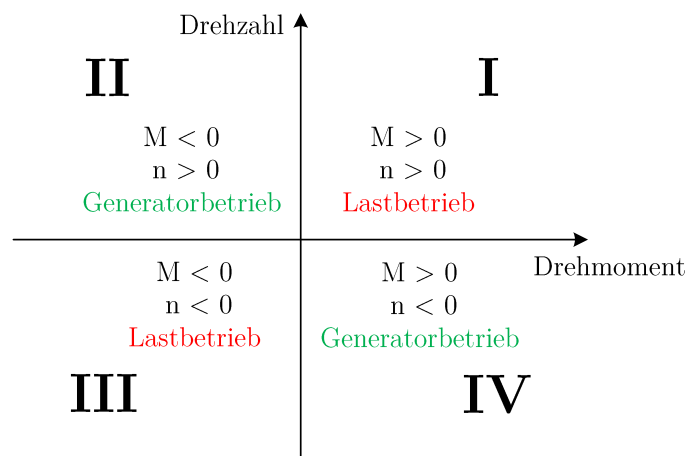


Abbildung 4.19: Quadranten und Vorzeichen der Motorkennlinie nach [47]

Zur Berechnung der Drehzahl im Lastbetrieb abhängig vom Drehmoment kann Formel 4.62 aus [47] verwendet werden. Hier muss zusätzlich berücksichtigt werden wann die maximale Leistung erreicht ist. Ist dies der Fall berechnet sich die Drehzahl mit Formel 4.63 für den Lastbetrieb und mit Formel 4.64 für den Generatorbetrieb nach [47]. Der Generatorbetrieb kann auch als Rekuperation bezeichnet werden und entspricht der Rückgewinnung an Energie [47].

$$n_{Mot} = n_{0\ Mot} - M_{Mot} \cdot k_{M\ Mot} \quad (4.62)$$

$$n_{Mot} = \frac{P_{max\ Mot}}{M_{Mot} \cdot 2 \cdot \pi} \quad (4.63)$$

$$n_{Mot} = \frac{P_{max\ rcu\ Mot}}{M_{Mot} \cdot 2 \cdot \pi} \quad (4.64)$$

mit:	n_{Mot}	: Drehzahl des Motors [1/s]
	$n_{0\ Mot}$: Leerlaufdrehzahl [1/s]
	M_{Mot}	: Drehmoment des Motors [Nm]
	$P_{max\ Mot}$: max. Leistung des Motors [W]
	$P_{max\ rcu\ Mot}$: max. Leistung des Motors bei Rekuperation [W]

Mit den dargelegten Formeln lässt sich das Motorkennfeld in Abbildung 4.20 bestimmen. Darin sind die Unterschiede des Betriebes bzw. bei Erreichung der max. Leistung an den Verlaufszweigen markiert.

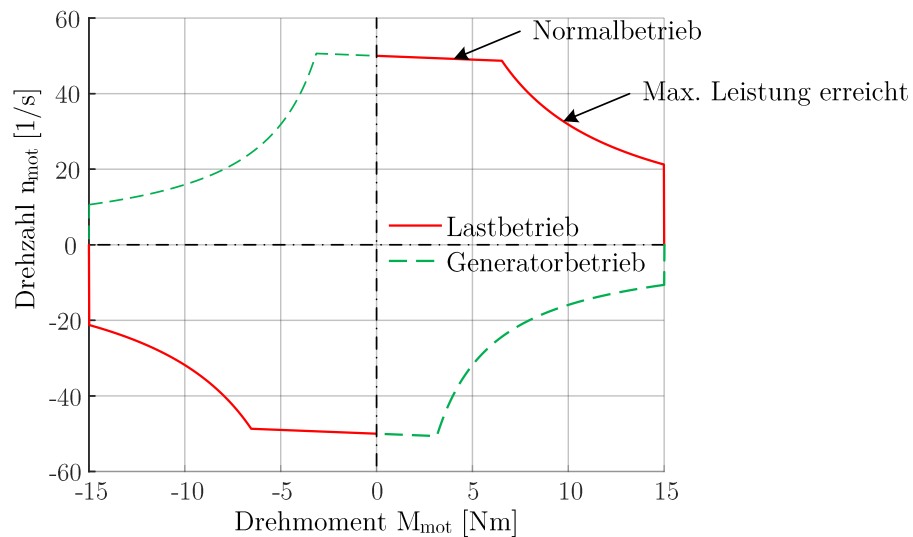


Abbildung 4.20: Motorkennlinie des Flugzeuges HAP

Bestimmung des Betriebspunktes

Im Motorkennfeld ist der Betriebspunkt zu ermitteln. Dieser muss sich innerhalb der Motorgrenzen befinden und ergibt sich durch den geforderten Schub auf der Propellerkennlinie. Zuletzt genannte wird ebenfalls in Abhängigkeit von Drehzahl und -moment angegeben und wurde zuvor bereits beschrieben. Befindet sich die geforderte Leistung außerhalb des Kennfeldes kann nur der max. Betriebspunkt verwendet werden. Folglich ist eine Lastreduzierung notwendig. In Abbildung 4.21 ist dies dargestellt für eine Fluggeschwindigkeit von 9 m/s EAS.

Die elektrische Leistung kann mit Formel 4.65 aus [47] berechnet werden. Hier werden die Drehzahl und das -moment des ermittelten Betriebspunktes eingesetzt. Die elektrische Leistung ergibt sich durch Formel 4.66 durch Verrechnung mit dem Wirkungsgrad. Es ist anzumerken, dass die Anzahl der Antriebe für die gesamte elektrische Leistung nach Formel 4.67 berücksichtigt werden muss.

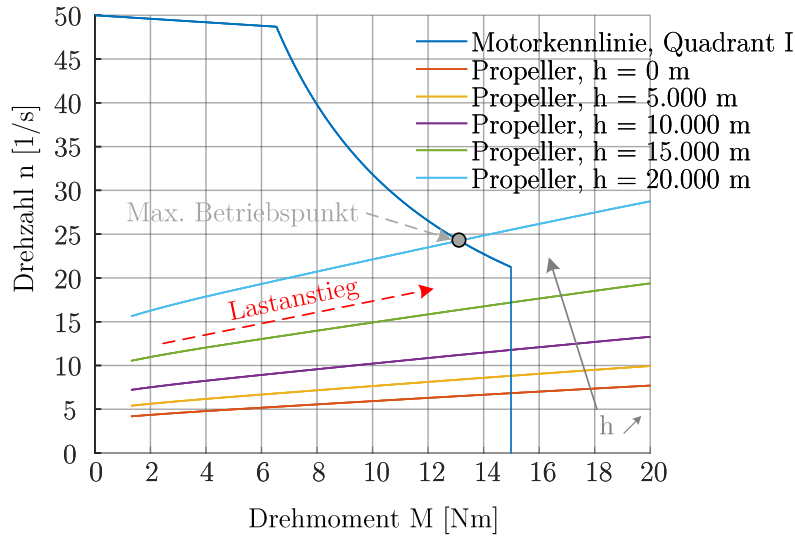


Abbildung 4.21: Betriebspunkt des Motors abhängig von der Flughöhe

$$P_{shaft\ Mot} = 2 \cdot \pi \cdot n_{Mot} \cdot M_{Mot} \quad (4.65)$$

$$P_{el\ Mot} = \frac{P_{shaft\ Mot}}{\eta_{el\ Mot}} \quad (4.66)$$

$$P_{el\ tot\ Mot} = \sum_{i=1}^n P_{el\ Mot\ i} \quad (4.67)$$

mit: $P_{shaft\ Mot}$: Leistung der Abtriebswelle [W]
 n_{Mot} : Drehzahl des Motors [1/s]
 M_{Mot} : Drehmoment des Motors [Nm]
 $P_{el\ Mot}$: elektr. Leistung des Motors [W]
 $P_{el\ tot\ Mot}$: gesamte elektr. Leistung der Antriebe [W]
 $\eta_{el\ Mot}$: elektr. Wirkungsgrad [-]

4.2.8 Solargenerator

Der Solargenerator wird beschrieben durch viele einzelne Solarzellen, die auf der Flügeloberseite tiefen- als auch spannweitenseitig verteilt sind. Zur Berechnung der zugeführten Gesamtenergie durch die Sonne sind verschiedene Koordinatentransformationen notwendig.

Zuerst muss die direkte Sonnenstrahlung ε_{dir} , angegeben im geodätischen KOS, in das flugzeugfeste KOS transformiert werden. Der Anteil von ε_{dir} ist in Formel 4.68 durch den Vektor p_{sun} dargestellt. p_{sun} wird im Folgenden als die spezifische Sonnenenergie bezeichnet. Dies dient lediglich der korrekten mathematischen Beschreibung.

$$p_{sun} = \begin{pmatrix} \varepsilon_{dir} \\ 0 \\ 0 \end{pmatrix} \quad (4.68)$$

Die in Kapitel 4.2.1 bereits beschriebene Transformationsmatrix T_{fg} nach Formel 4.16 wird für die Transformation benötigt. Weiterhin ist eine Drehung um die y-Achse mit dem Höhenwinkel α_S erforderlich. Dieser lässt sich mit Formel 4.69 aus dem Zenitwinkel ξ berechnen.

$$\alpha_s = \frac{\pi}{2} - \xi \quad (4.69)$$

Die gesamte Drehreihenfolge zur Ermittlung der spezifischen Sonnenenergie im flugzeugfesten KOS ergibt sich in Formel 4.70. Die ausführlichen Drehmatrizen folgen darunter.

$$(p_{sun})_f = T_{x\phi_{roll}} \cdot T_{y\theta} \cdot T_{z\Psi_{delta}} \cdot T_{y\alpha_s} \cdot p_{sun} \quad (4.70)$$

mit: $(p_{sun})_f$: spezifische Sonnenstrahlung im flugzeugfesten KOS [W/m^2]
 $T_{x\phi_{roll}}$: Drehmatrix um die X-Achse mit dem Winkel ϕ_{roll}
 $T_{y\theta}$: Drehmatrix um die Y-Achse mit dem Winkel θ
 $T_{z\Psi_{delta}}$: Drehmatrix um die Z-Achse mit dem Winkel Ψ_{delta}
 $T_{y\alpha_s}$: Drehmatrix um die Y-Achse mit dem Winkel α_s
 p_{sun} : spezifische Sonnenstrahlung im geodätischen KOS [W/m^2]

$$T_{x\phi_{roll}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_{roll} & \sin \phi_{roll} \\ 0 & -\sin \phi_{roll} & \cos \phi_{roll} \end{pmatrix} \quad (4.71)$$

$$T_{y\theta} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (4.72)$$

$$T_{z\Psi_{delta}} = \begin{pmatrix} \cos \Psi_{delta} & \sin \Psi_{delta} & 0 \\ -\sin \Psi_{delta} & \cos \Psi_{delta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.73)$$

$$T_{y\alpha_s} = \begin{pmatrix} \cos \alpha_s & 0 & -\sin \alpha_s \\ 0 & 1 & 0 \\ \sin \alpha_s & 0 & \cos \alpha_s \end{pmatrix} \quad (4.74)$$

Der Winkel Ψ_{delta} ergibt sich nach Formel 4.75 und beschreibt die Differenz des Azimut des Flugzeuges zum Azimut der Sonne.

$$\Psi_{delta} = \Psi - \Psi_S \quad (4.75)$$

Da das individuelle Solarpanel eine eigene Orientierung relativ zum Flugzeug besitzt, erhält jedes Solarpanel seine eigenen Orientierungswinkel, schematisch dargestellt in Abbildung 4.22. Für jedes Panel ist daher eine individuelle Koordinatentransformation notwendig nach Formel 4.76. Die zugehörigen Drehmatrizen sind in Formel 4.77 und 4.78 gezeigt.

$$(p_{sun})_{pi} = T_{x\phi_{SP}} \cdot T_{y\theta_{SP}} \cdot (p_{sun})_f \quad (4.76)$$

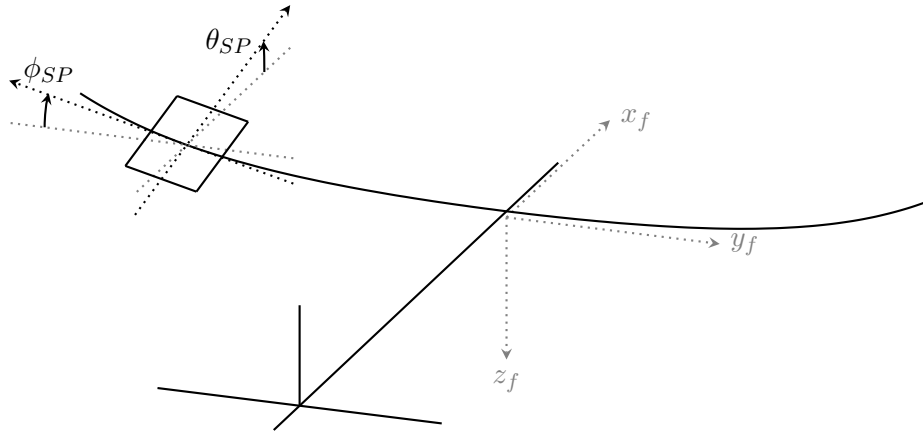


Abbildung 4.22: Winkel der Solarpaneele am Flugzeug

$$T_{x\phi_{SP}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_{SP} & \sin \phi_{SP} \\ 0 & -\sin \phi_{SP} & \cos \phi_{SP} \end{pmatrix} \quad (4.77)$$

$$T_{y\theta_{SP}} = \begin{pmatrix} \cos \theta_{SP} & 0 & -\sin \theta_{SP} \\ 0 & 1 & 0 \\ \sin \theta_{SP} & 0 & \cos \theta_{SP} \end{pmatrix} \quad (4.78)$$

- mit: $(p_{sun})_{pi}$: spezifische Sonnenstrahlung im individuellen Panel KOS [W/m^2]
 $T_{x\phi_{SP}}$: Drehmatrix um die X-Achse mit dem Winkel ϕ_{SP}
 ϕ_{SP} : Winkel der Panelorientierung um die X-Achse
 $T_{y\theta_{SP}}$: Drehmatrix um die Y-Achse mit dem Winkel θ_{SP}
 θ_{SP} : Winkel der Panelorientierung um die Y-Achse

Die zu verwendende spezifische Sonnenstrahlung des individuellen Panels ergibt sich aus der dritten Komponente des erhaltenen Vektors $(p_{sun})_{pi}$. Dieser Zusammenhang ist in Formel 4.79 dargelegt.

$$p_{sun\ i} = (p_{sun})_{pi} \quad (3) \quad (4.79)$$

- mit: $p_{sun\ i}$: spezifische Sonnenstrahlung des individuellen Panels [W/m^2]

Damit kann die erhaltene Energie eines Solarpanels mit Formel 4.80 berechnet werden. Die Summe aller Panelenergien ergibt nach Formel 4.81 die Gesamtenergie des Solargenerators.

$$P_{SP\ i} = p_{sun\ i} \cdot A_{SP\ i} \cdot \eta_{SP\ i} \quad (4.80)$$

$$P_{SG} = P_{SP\ i} + P_{SP\ i+1} + \dots \quad (4.81)$$

- mit: P_{SG} : Energie des Solargenerators [W]
 $P_{SP\ i}$: Energie des Solarpanels i [W]
 $A_{SP\ i}$: Fläche des Solarpanels i [m^2]
 $\eta_{SP\ i}$: Wirkungsgrad des Solarpanels i [-]

Es ist anzumerken, dass jedes Solarpanel mit einem eigenen Wirkungsgrad modelliert ist. Der Gesamtwirkungsgrad des Solargenerators lässt sich mit Formel 4.82 berechnen.

$$\eta_{SG} = \eta_{SPi} \cdot \frac{A_{SPi}}{A_{SG}} + \eta_{SPi+1} \cdot \frac{A_{SPi+1}}{A_{SG}} + \dots \quad (4.82)$$

mit: η_{SG} : Gesamtwirkungsgrad des Solargenerators [-]
 A_{SG} : Gesamtfläche des Solargenerators [m^2]

Einflüsse auf das individuelle Solarpanel durch beispielsweise Abschattung wurden nicht modelliert.

4.2.9 Batteriemodell

Das Batteriesystem besteht aus mehreren Batterieblöcken, die jeweils im Flügelholm untergebracht sind. Zur Beschreibung dieser Batterien in ihrer Gesamtheit wird die Batteriemasse und die Energiedichte verwendet. Die zur Verfügung stehende elektrische Energie ergibt sich dadurch nach Formel 4.83. Der darin enthaltene State of Charge (SoC) gibt den Ladestand des Batteriesystems an. Ein Ladestand von 0 entspricht kompletter Entladung. Entsprechend gibt ein SoC von 1 die volle Ladung der verfügbaren Gesamtkapazität an. Jedoch wird diese durch einen Degradierungsfaktor und eine Temperaturabhängigkeit skaliert nach den Modellen aus [48] [49]. (Anmerkung: Formelparameter in 4.83, die separat berechnet sind, werden anschließend beschrieben)

$$E = m_{bat} \cdot e_{bat} \cdot SoC \cdot \zeta_{dgr} \quad (4.83)$$

mit: E : elektrische Energie [Ws]
 m_{bat} : Batteriemasse [kg]
 e_{bat} : Energiedichte [Ws/kg]
 SoC : State of Charge [-]
 ζ_{dgr} : Degradierungsfaktor [-]

Die Energiedichte e_{bat} berechnet sich weiter nach Formel 4.84. Die darin enthaltene Skalierung berücksichtigt die Temperaturabhängigkeit der Energiedichte. Diese wird mithilfe eines Kennfeldes bestimmt, dargestellt in Abbildung 4.23. Darin wird die maximale Energiedichte abhängig von der Batterietemperatur angegeben. Die Batterien werden diesbezüglich dauerhaft auf eine vordefinierte Temperatur geheizt. Die Daten zum Kennfeld stammen aus einem Thermalmodell von [1].

$$e_{bat} = \frac{e_{batT}}{e_{batT25}} \cdot e_{bat0} \quad (4.84)$$

mit: e_{batT} : Energiedichte bei Temperatur T [Ws/kg]
 $e_{batT25} = 1$: Energiedichte bei Temperatur $T = 25^\circ C$ [Ws/kg]
 e_{bat0} : Theor. mögliche Energiedichte [Ws/kg]

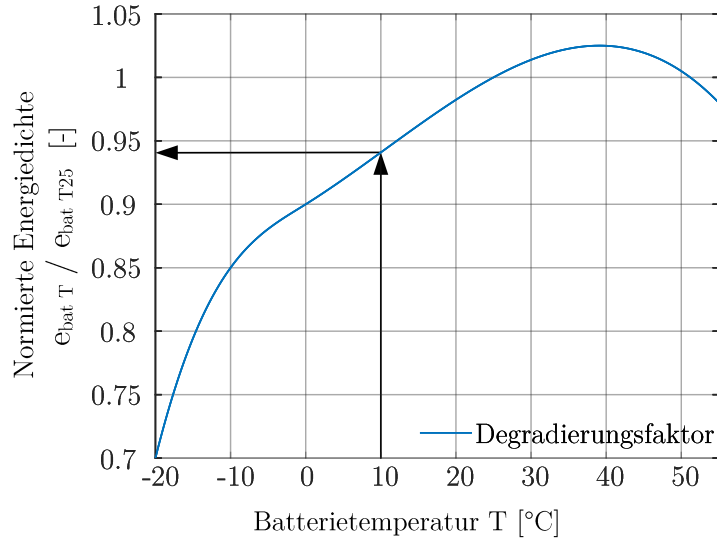


Abbildung 4.23: Max. Batteriekapazität abhängig von der Batterietemperatur

Der Degradierungsfaktor ζ_{dgr} beschreibt den Anfangszustand der Batterien sowie die Degradierung nach entsprechender Anzahl an Be- und Entladezyklen. Er wird berechnet nach Formel 4.85 und 4.86. Der darin enthaltene Faktor σ_n beschreibt den Kapazitätsverlust nach einer bestimmten Anzahl an Zyklen voller Be- und Entladung. Nach [49] wird ein Wert von 2,5 % pro 200 Zyklen angenommen, der in Formel 4.86 bereits verrechnet ist.

$$\zeta_{dgr} = \zeta_{dgr 0} \cdot \zeta_{dgr n} \quad (4.85)$$

$$\zeta_{dgr n} = 1 - \left(\frac{1}{200} \cdot \sigma_n \cdot \sum_{i=1}^{n_{sim}} n_{chgi} \right) \quad (4.86)$$

mit: $\zeta_{dgr 0}$: Degradierungsfaktor bei Missionsstart [-]
 $\zeta_{dgr n}$: Degradierungsfaktor nach n_{sim} Zyklen [-]
 $\sigma_n = 0,0025$: Degradierungsrate [-]
 n_{chgi} : Äquivalenter Ladezyklus i [-]
 n_{sim} : Nummer der Simulationsschleife [-]

Da während der Simulation die Batterien nicht vollständig Be- und Entladen werden, wird ein äquivalenter Ladezyklus in Formel 4.87 nach [48] berechnet. Dieser wird während jedes Durchlaufes neu berechnet und entsprechend in Formel 4.86 aufsummiert.

$$n_{chgi} = \frac{|E_i|}{m_{bat} \cdot e_{bat 0}} \quad (4.87)$$

mit: E_i : Energieladung oder -verbrauch eines Simulationsschrittes [Ws]

Die Heizleistung der Batterien ist abhängig von der zu heizenden Temperatur, der aktuellen Flughöhe und der Tageszeit. Die Flughöhe ist relevant, da die Luftdichte mit steigender Höhe abnimmt.

Folglich nimmt der konvektive Wärmegang stark ab. Die Tageszeit spiegelt den Sonnenstand bzw. die verfügbare Wärme durch die Sonnenstrahlung wieder. Die Heizleistung wird mittels der spezifischen Heizleistung interpoliert. Diese ist in Abbildung 4.24 pro 1,3 kg Batteriemasse nach [1] dargestellt. Weiterhin wird abhängig von der Tageszeit die entsprechende Leistung entnommen. Die Heizleistung ergibt sich nach Formel 4.88. Die Heizenergie kann mit Formel 4.89 berechnet werden.

$$P_{heat} = 1,3 \cdot p_{heat} \cdot m \quad (4.88)$$

$$E_{heat} = P_{heat} \cdot dt \quad (4.89)$$

mit: E_{heat} : Energie zum Heizen der Batterien [Ws]
 P_{heat} : Heizleistung [W]
 p_{heat} : spezifische Heizleistung [W/kg]
 dt : Zeitabstand eines Simulationsschrittes [s]

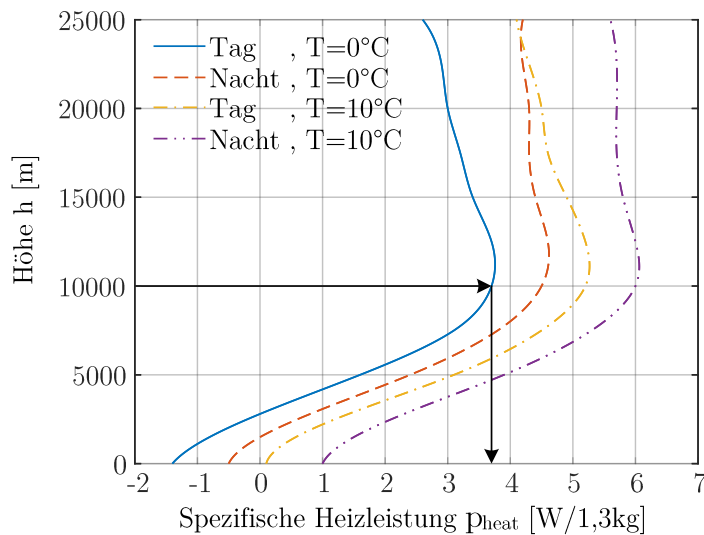


Abbildung 4.24: Spezifische Heizleistung abhängig von Tageszeit und Höhe

Zum aktuellen SoC wird noch der SoC_0 und der SoC_{min} verwendet. Der erst genannte SoC_0 dient zur Initialisierung der Startkonfiguration und soll somit Zwischenladestände abbilden. Der Parameter SoC_{min} dient dem Schutz der Batterien vor zu tiefer Entladung und zur Minimalversorgung der Bordsysteme.

5 Verifikation der Simulation

Die Verifikation der verwendeten Methoden und Formeln der dargelegten Simulation ist von besonderer Bedeutung, da Fehler oder falsche Annahmen signifikante Auswirkungen auf das Simulationsergebnis haben können. Dies ist mitunter zu begründen durch die Kumulation eines hypothetischen Fehlers aufgrund der sehr hohen Anzahl an Simulationsschritten. Beispielsweise lag die Anzahl der Simulationsschritte während der Ausarbeitung im Bereich von 2.000 bis 10.000 Simulationsrunden. Darüber hinaus kann die Fehlerausbreitung je nach Parameter innerhalb der Simulation linear, quadratisch, kubisch und exponentiell erfolgen. Ein Fehler innerhalb einer Formel oder des Ablaufes des Programms ist bei Änderungen und Anpassungen leicht zu übersehen bzw. er könnte sogar in den Ergebnissen übersehen werden. Folglich ist die kontinuierliche Verifikation der Simulation anzustreben und zu empfehlen.

5.1 Methoden zur Verifizierung

Im Folgenden werden verschiedene Methoden vorgestellt mit denen das entwickelte Simulationsprogramm auf eine korrekte Funktionsweise und Ausgabe der Ergebnisse überprüft wird.

5.1.1 Flugabschnitte

Berechnungen und Ergebnisse einzelner Flugabschnitte bzw. Missionssegmente können **nicht-formal** verifiziert werden. Darunter versteht man nach [50] das dynamische und statische Testen des Programms um entstandene Fehler während der Programmentwicklung zu eliminieren.

Das **dynamische Testen** ist charakterisiert durch die Ausführung von Testfällen. Dazu besteht die Auswahl einer Testmission innerhalb der Eingabestruktur. Bei Ausführung dieser Testmission kann der Nutzer selbst oder automatisch Flugabschnitte simulieren lassen. Zur Übersichtlichkeit werden dabei nur ca. 3-20 Simulationsschritte je nach Auswahl ausgeführt. Die so entstandenen Ergebnisse können mit analytisch bestimmten Ergebnissen verglichen werden.

Beim **statischen Testen** werden einzelne Programmabschnitte analysiert. Dabei sind neben Schleifenfunktionen vor allem verschachtelte Bedingungsabfragen zu begutachten. Dies kann mit der Durchführung von Testmissionen gekoppelt werden durch Platzierung verschiedener Kontrollausgaben. Weiterhin sind die verwendeten Formeln nochmals zu überprüfen, beispielsweise durch Duplizieren der Formel in einen kleineren Testbereich.

Besonders bei der Verifizierung besticht die Objektorientierung, da die einzelnen Objekte unabhängig vom Gesamtsystem erstellt und getestet werden können. Die Objektfunktionen können durch für sie zugeschnittene Unittests verifiziert werden. Unter einem Unittest versteht man das abgegrenzte Testen einer Komponente zur Überprüfung von Zwischenergebnissen [51]. Als Beispiel ist in Abbildung 5.1 der Unittest für die Funktion `Aircraft.CalculateFlightState` dargestellt. Dabei werden der Funktion Eingabewerte für den Bahnneigungswinkel γ übergeben, welche sich im Bereich von $\pm 30^\circ$ befinden. Die Schrittweite liegt bei $0,1^\circ$. Verifiziert wird durch die Auswertung der Ausgaben.

Durch Automatisierung, beispielsweise durch Verwendung weiterer Unterprogramme, können Unittests automatisch durchgeführt werden. Jedoch sollte zeitweise eine manuelle Verifizierung durchgeführt werden zur zusätzlichen Absicherung. Dies resultiert daraus, dass vermeidlich korrekte Änderungen innerhalb der Unittests einen unentdeckten Fehler verdecken können.

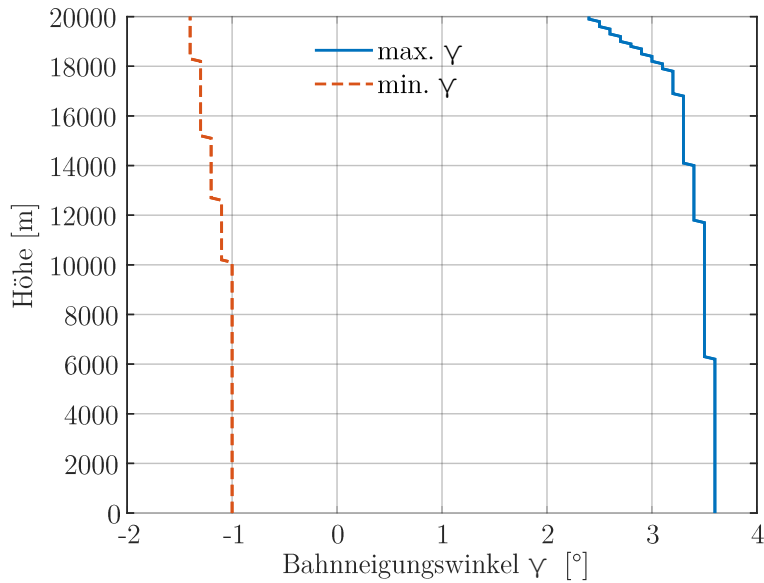


Abbildung 5.1: Unittest für Aircraft.CalculateFlightState

5.1.2 Solarmodell

Das Solarmodell lässt sich zusätzlich zu den vorgestellten Methoden durch vorhandene Solarmodelle aus externen Quellen verifizieren. Dabei werden die solare Strahlung und/oder die Sonnenstände bzw. der Verlauf der Sonnenwinkel verglichen. Zur Verifikation können verschiedene Eingangsparameter untersucht werden wie beispielsweise die Flughöhe, der Tag im Jahr und Positionen auf der Erde.

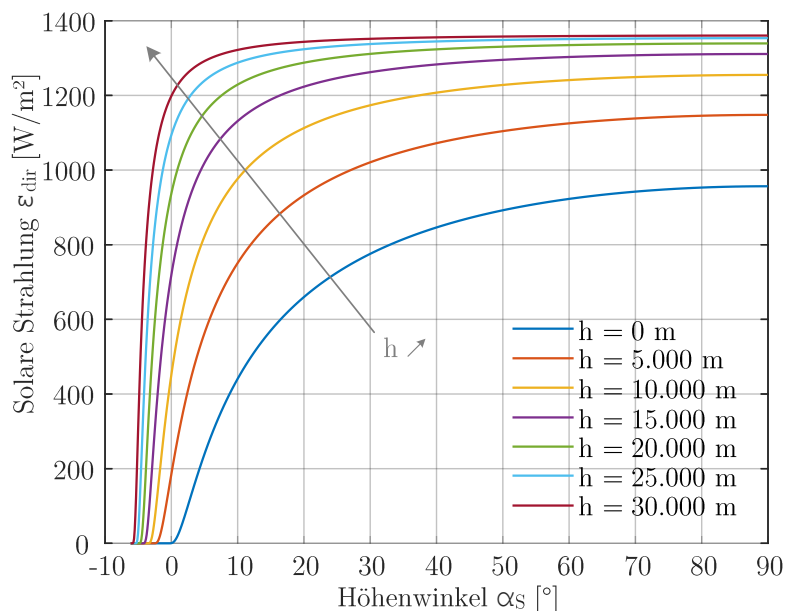


Abbildung 5.2: Solare Strahlung ϵ_{dir} in Abhängigkeit des Höhenwinkel α_S

Die solare Strahlung ϵ_{dir} lässt sich für verschiedene Flughöhen über den Höhenwinkel α_S mit [34] verifizieren. Die Ergebnisse sind in Abbildung 5.2 gezeigt und stimmen mit der Datenquelle überein.

Außerdem ist in der Abbildung der Einfluss der Depression ξ_{dep} mit steigender Höhe durch die Verschiebung zu negativen Höhenwinkeln ersichtlich.

In Abbildung 5.3 wird der Verlauf des Zenitwinkels bzw. Höhenwinkels aus [34] überprüft. Die Eingaben gelten für den Winter (=Tag 0) am Breitengrad 48° Nord in 20.000 m Flughöhe. Der Längengrad ist hier nicht relevant. Die Ergebnisse des Sonneverlaufs decken sich mit den Daten aus [34]. Der Verlauf des Azimut ist in Abbildung 5.4 gezeigt. Er beginnt plausibel bei -180° und verläuft über den Tagesverlauf wieder zu $+180^\circ$. Zur genaueren Aufschlüsselung bzw. detaillierteren Überprüfung können die Komponenten der Transformationsmatrix T_{ge} herangezogen werden. Diese und weitere Sonnenverläufe sind in Anhang A zu finden.

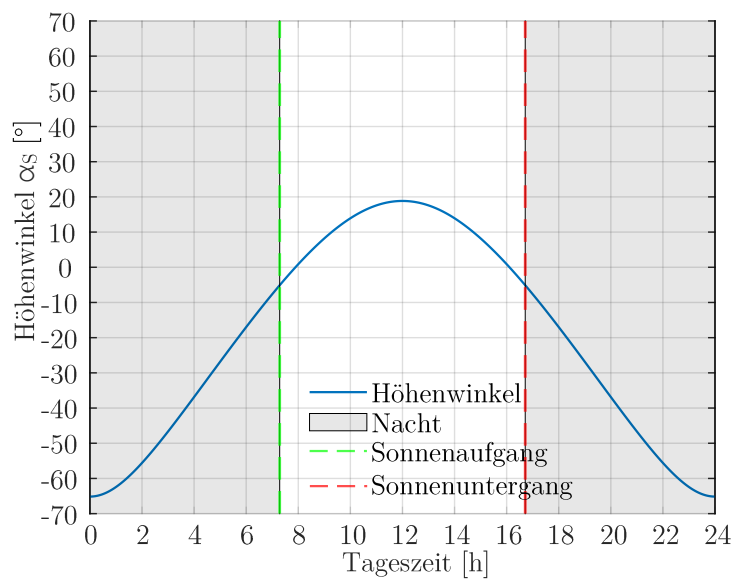


Abbildung 5.3: Sonnenverlauf im Winter für $\phi = 48^\circ$ N; $h = 20.000$ m

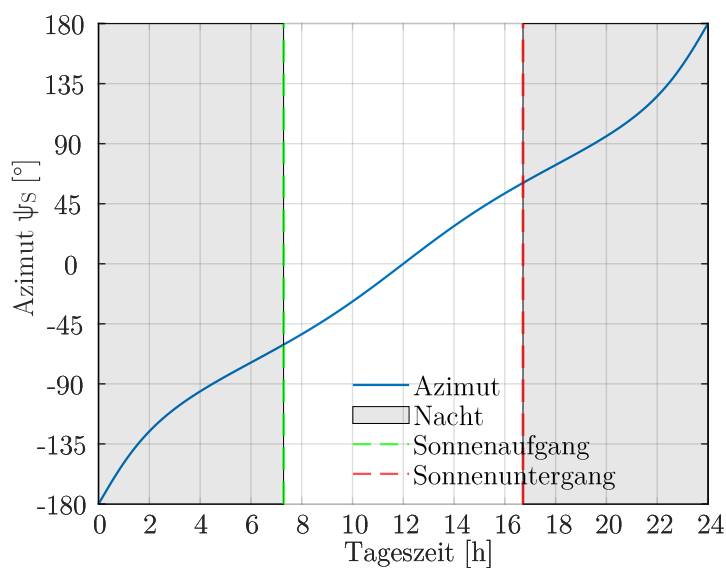


Abbildung 5.4: Azimutverlauf im Winter für $\phi = 48^\circ$ N; $h = 20.000$ m

5.1.3 Iterlogs

Die Verwendung des Attributs **Iterlog**, welches in Kapitel 3 vorgestellt wurde, ist als **Built-in Test** aufzufassen. Diese beschreiben Testbausteine, die während der Simulation aktiv verwendet werden [52]. Im konkreten Fall werden sie als Zähler interner Iterationsschleifen eingesetzt, die ebenfalls abgespeichert werden. Die Auswertung der Iterlog Parameter kann automatisch durch Grenzwerte oder durch den Nutzer erfolgen. Sind beispielsweise extrem hohe oder unschlüssige Iterationsdurchläufe notwendig, kann dies ein Indikator für einen vorhandenen Fehler sein. Eine Auswertung der Iterlog Parameter ist in Abbildung 5.5 gezeigt. In diesem Beispiel lassen sich auch Flugabschnitte erkennen wie beispielsweise der Übergang zum Horizontalflug. Weiterhin ist erkennbar, dass zum Simulationsende die Batterie entladen ist, weshalb der entsprechende Iterlog anschlägt. Jedoch ist auch eine Anomalie in Iterlog.motor erkennbar. Der konstante Ausschlag zeigt, dass innerhalb dieser Funktion ein interner Fehler vorliegt. Dieser verursacht neben einer Reduktion der Flugleistung eine Erhöhung der Rechenzeit aufgrund einer konstant vorhandenen Iterationsschleife pro Simulationsschritt. Da in diesem Fall lediglich eine zusätzliche Iteration pro Durchlauf auftritt, kann auf einen falsch platzierten Marker geschlossen werden.

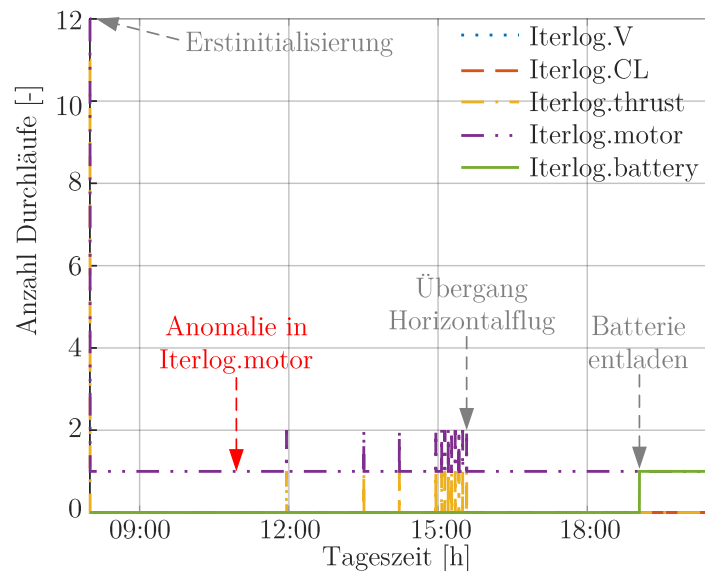


Abbildung 5.5: Iterlog Parameter verschiedener Objektfunktionen

5.2 Verifikation durch HALE Flugzeuge

Eine weitere Methode zur Verifizierung der Simulationsergebnisse ist die Simulation bereits vorhandener Solarflugzeuge wie beispielsweise die vorgestellten HALE-Flugzeuge aus Kapitel 2. Dabei wurden vorhandene Datensätze zu den Flugzeugen BAE PHASA 35 und Zephyr S untersucht.

Zur Verifizierung durch die gewählten Flugzeuge sind ausreichende Daten zum Flugzeug selbst sowie den Missionsverläufen und -daten notwendig. Erst genannte Daten zum Flugzeug sind meist in Veröffentlichungen oder auf den Webseiten der Hersteller zu finden. Jedoch sind speziell zu den Missionsverläufen keine öffentlich zugänglichen Daten vorhanden, die genügend Detailgrad aufweisen. Diese Problematik versperrt folglich diese Verifizierungsmethode im vorliegenden Fall.

Allerdings ist es möglich Simulationen mit den Flugzeugen durchzuführen. Dadurch können Leistungsdaten abgeschätzt werden auf Basis der Genauigkeit der Eingangsdaten. Als Beispiel kann hier die Steigleistung genannt werden. Bezüglich der Analyse sind abhängig von den vorhandenen Daten zwei Verfahren möglich, die aus [42] entnommen und für die Missionssimulation abgeleitet wurden. So kann je nach Datenzugang nach **Entwurfs-** und **Nachrechensimulation** unterschieden werden.

Die **Entwurfssimulation** definiert sich durch vorhandene Missionsdaten und/oder -profile aus denen die Komponenten des Flugzeuges ermittelt werden müssen. Zur Unterstützung hilft es einige wenige Kenndaten des Flugzeuges zu kennen. Dies erlaubt die Ermittlung von Schlüsselparametern wie der Energiedichte und Masse der Batterien als Beispiele. Die **Nachrechensimulation** bildet die klassische Simulation ab, bei der das Flugzeug komplett beschrieben ist und die Missionsergebnisse zu ermitteln sind.

6 Simulationsergebnisse zum Projekt HAP

Im Folgenden werden die Ergebnisse und Erkenntnisse der Simulation vorgestellt und beschrieben. Dabei werden zuerst Stellgrößen innerhalb der Simulation und ihr Einfluss vorgestellt. Die daraus optimierte Simulationseinstellung wird für die Durchführung der Standardmission verwendet. Die Bezeichnung Standardmission dient lediglich der eindeutigen Zuweisung innerhalb dieser Dokumentation. Neben den verwendeten Eingabeparametern werden die Ergebnisse analysiert und Schlussfolgerungen gegeben. Abschließend wird die Erweiterung der Simulation für Studien vorgestellt.

6.1 Stellgrößen der Simulation

Die für den Ablauf der Simulation relevanten Stellgrößen sind im Feld des Attributes **Scenario.iter** hinterlegt. Für die zu simulierende Mission wurden die Werte in Tabelle 6.1 gesetzt. Die Beschreibung der Parameter ist in Tabelle 3.1 (siehe Kapitel 3) nachzulesen. Die Wahl für die simulationsbeeinflussenden Werte, in Tabelle 6.1 mit * gekennzeichnet, basieren auf Erkenntnissen während der Entwicklung. Einige dieser werden nachfolgend erläutert. Es ist anzumerken, dass die Einflüsse mit reduzierter Anzahl max. Simulationsrunden ermittelt wurden.

Parameter	Wert	Einheit	Bemerkung
dt	10	s	*
ds	0,1	m/s	*
da	0,1 · $\pi/180$	rad	*
sim_round	1	–	
sim_maxrounds	10.000	–	*
threshold	0.005	–	*
counter_max	100	–	
checkpoint	30	–	
delta_0	0,01	–	
counter_0	1	–	
overload	1	–	

(* $\hat{=}$ Stellgröße)

Tabelle 6.1: Gesetzte Parameter des Arrays iter

Der Parameter **iter.dt** steuert die Schrittweite eines Simulationsschrittes und beeinflusst damit neben der Genauigkeit einer betrachteten Flugdistanz auch die max. Missionsdauer. Letzteres resultiert aus der Kopplung mit **iter.sim_maxrounds**. Die Auswirkung verschiedener Werte für dt ist in Abbildung 6.1 dargestellt. Daraus folgt, dass der Nutzer die Anzahl an max. Simulationsrunden abhängig seines Analyse- und/oder Missionszieles anpassen muss.

Der Einfluss der Steuergröße **iter.ds** wird nicht betrachtet, da dieser nur angewendet wird bei Überschreitung der Flugbereichsgrenzen. Da dies jedoch bei korrekt gewählten Missionsdaten entfällt, wird der Einfluss nicht beschrieben.

Demgegenüber findet der Parameter **iter.da** häufig Anwendung, da als zentrale Steuergröße des Fluges der Bahnneigungswinkel γ verwendet wird. Dieser wird bei gesetztem Overload Marker durch iter.da im Steig- oder Reiseflug reduziert bzw. im Sinkflug erhöht. Sein Einfluss auf den Flugpfad ist in Abbildung 6.2 gezeigt. In den Ergebnissen wird verdeutlicht, dass sich neben dem Missionsende auch andere Parameter verändern. Als Beispiel dazu lässt sich aus Abbildung 6.2 die Steigrate herauslesen. Nach [1] ist ein Wert von $0,1^\circ$ für iter.da technisch realistisch umsetzbar.

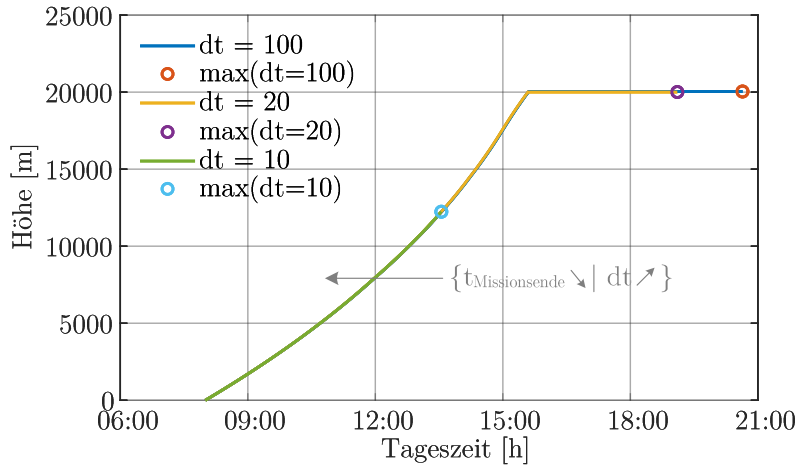


Abbildung 6.1: Einfluss des Parameters iter.dt auf den Missionsverlauf

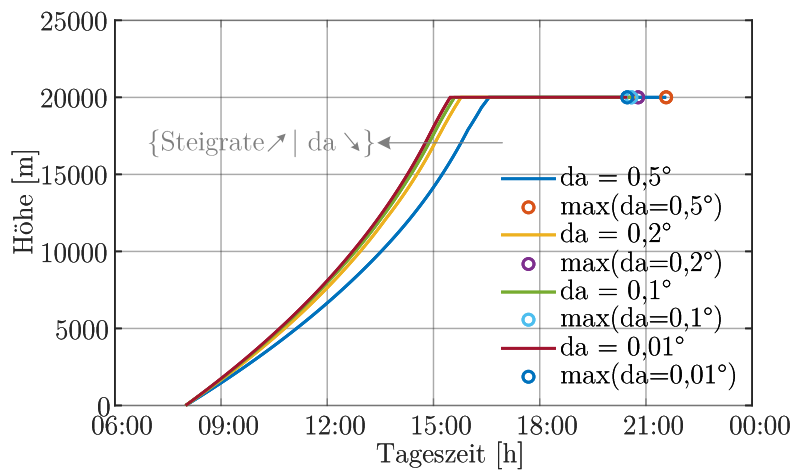


Abbildung 6.2: Einfluss des Parameters iter.da auf den Missionsverlauf

Der Parameter `iter.threshold` dient als Schwellenwert zum Vergleich der Werte der aktuellen Iteration mit dem der vorangegangenen Iteration. Typischerweise wird er in Prozent angegeben. Sein Einfluss auf die Mission zeigt sich in Abbildung 6.3 durch eine Reduktion der Steigleistung resultierend aus höheren Leistungsdaten durch ein vorzeitiges Iterationsende. Daraus folgt, dass sich das Flugzeug bei größeren Werten von `iter.threshold` näher an seinen Leistungsgrenzen befindet.

Die Anzahl der max. Simulationsrunden `sim_maxrounds` begrenzt die Missionszeit abhängig von `iter.dt`. Daher ist eine geeignete Auswahl für ihn zu wählen, idealerweise so gesetzt, dass die Mission gerade noch erfüllt wird. Ein zu hoher Wert belastet die Speicherbereitstellung und kann je nach Leistung der verwendeten Hardware die Rechenzeit beeinflussen.

Basierend auf den bisherigen Erkenntnissen ist die Verfeinerung der Parameter sinnvoll. Jedoch sollte dabei auch die Simulationszeit beachtet werden, da durch feinere Einstellungen mehr Simulations- und Iterationsrunden durchlaufen werden müssen. Zusätzlich zu den dargelegten Stellgrößen sind auch die Methoden der Iteration bezüglich der Simulationszeit zu betrachten. Untersucht wurden das **äquidistante** und das **Halb-Schritt** Iterationsverfahren nach [53] sowie die Kopplung durch die Funktion `Mission.Optimizer`.

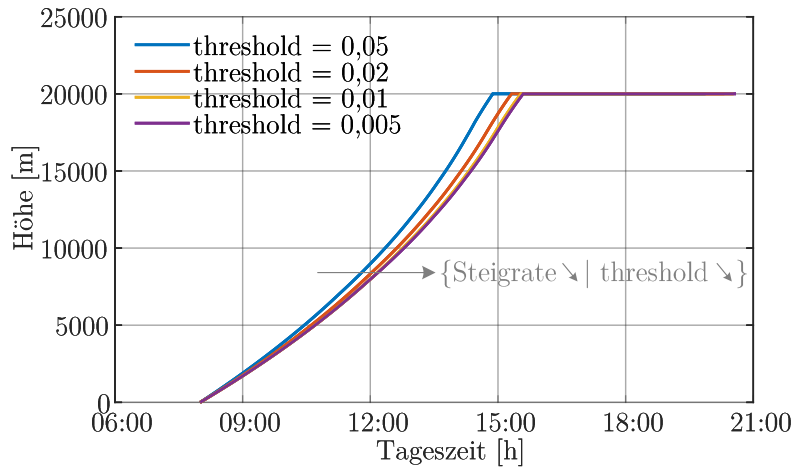


Abbildung 6.3: Einfluss des Parameters iter.threshold auf den Missionsverlauf

Beim äquidistanten Verfahren ist die Schrittweite zur Lastreduktion fest vorgegeben. Das bedeutet, wenn der Zielwert weit vom Startpunkt entfernt ist, müssen viele Iterationen durchlaufen werden. Die Halb-Schritt Iteration verwendet innerhalb einer Iteration den halben Wert des Intervalls zwischen dem aktuellen Wert und dem Minimalwert des gesuchten Bereiches. Eine Auswertung der dadurch erzeugten Bereiche ergibt die weitere Laufrichtung der Folgeiterationen. Dieses Verfahren ist sehr dynamisch. Beide Verfahren iterieren in jedem Simulationsschritt erneut aufgrund der von der Mission geforderten Flugleistungen. Die Funktion Mission.Optimizer vergleicht die geforderten Flugleistungen mit den möglichen des aktuellen Simulationsschrittes und bestimmt einen Differenzwert. Dieser wird beim nächsten Schleifendurchlauf verrechnet und reduziert entsprechend die angeforderte Flugleistung sofern der gleiche Flugabschnitt vorliegt.

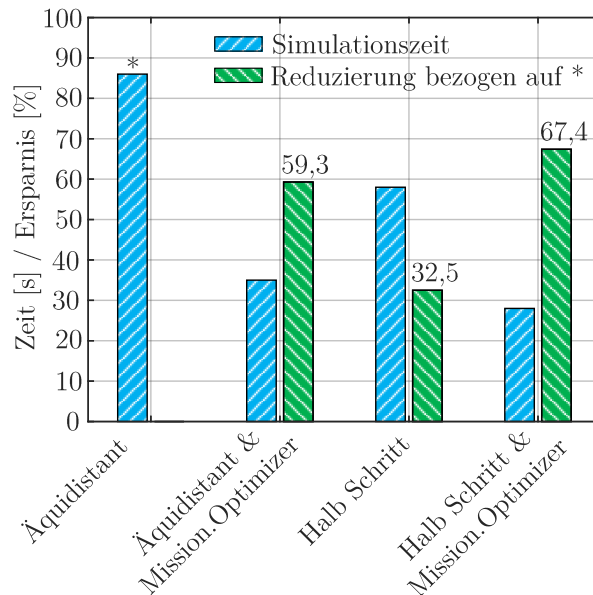


Abbildung 6.4: Einfluss der Iterationsmethoden auf die Simulationszeit

In Abbildung 6.4 sind Simulationszeiten abhängig des gewählten Iterationsverfahren dargestellt. Die Kombination aus Halb-Schritt Verfahren und Mission.Optimizer wird aufgrund der kürzesten Simula-

tionszeit gewählt. Diese Optimierung findet primär im Steigflug ihre Anwendung.

Zusätzlich kann die Simulationszeit weiter optimiert werden durch die Erstinitialisierung sämtlicher Variablen, welche auch Drehmatrizen umfassen. Dabei ist die Zeit zur Speicherbereitstellung abhängig von der Variablengröße und -art. Beispielsweise konnte die Platzierung der fertigen Transformationsmatrix T_{ge} zum Zeitpunkt der Umsetzung eine Ersparnis von 33,7 % der Simulationszeit bewirken. Dies beinhaltet auch die Einsparung durch die dadurch ersetzten Matrixmultiplikationen der Einzeldrehmatrizen. Nachteilig daran ist, dass sich die Nachvollziehbarkeit im Programm erschwert.

6.2 Standardmission

6.2.1 Eingabeparameter

Zur Durchführung der Standardmission sind Eingaben zum Flugzeug, der Mission und zum Windmodell erforderlich. Bezüglich des Flugzeuges werden in Tabelle 6.2 einige Schlüsselparameter zum Flugzeug HAP aufgeführt. Diese werden neben der Heiztemperatur der Batterien für die Standardmission als konstant behandelt. Eine Übersicht aller Daten der Standardmission ist in Anhang A gegeben. Die Mission wird mit Mission_0 indiziert zur eindeutigen Kennzeichnung. Sie besteht aus einem Steigflug mit anschließendem Horizontalflug auf Reisehöhe. Beide Segmente werden kreisend ausgeführt. Die entsprechenden Daten der Startzeit und -ort sind in Tabelle 6.3 gelistet. Die Bahndaten der Segmente sind in Tabelle 6.4 aufgeführt. Es wird bewusst auf ein Segment mit Sinkflug verzichtet, da vorrangig die Stationierbarkeit des Flugzeuges untersucht werden soll.

Parameter	Kürzel	Wert	Einheit
Flugzeugmasse	m	138	kg
Flügelfläche	S	35,9	m ²
Batteriemasse	m_{bat}	50	kg
Energiedichte Batterien	e_{bat}	210	Wh/kg
Heiztemperatur Batterien	T_{heat}	10	°C
Fläche Solar Generator	A_{SG}	10	m ²
Wirkungsgrad Solar Generator	η_{SG}	0,27	–
Flügelprofil		MH139R	
Höhenleitwerksprofil		NACA0009	
Propellerprofil		MH2	
Motor		MOTHAP	

Tabelle 6.2: Schlüsselparameter des Flugzeuges HAP

Die Eingaben des Windmodells entsprechen dem Standort Kiruna. Die dazugehörigen Daten sind in Kapitel 4 Abbildung 4.9 aufgezeigt. Es ist anzumerken, dass für die Standardmission die Windgeschwindigkeiten nicht mit den Fluggeschwindigkeiten verrechnet werden, da der Einfluss des Windes in Folgeanalysen außerhalb dieser Arbeit untersucht werden soll.

Parameter	Kürzel	Wert	Einheit
Startuhrzeit	t_0	8	h
Starttag	n_0	142	—
Breitengrad	ϕ	32	$^\circ$
Längengrad	λ	0	$^\circ$

Tabelle 6.3: Zeit- und ortbezogene Daten zur Mission_0

Parameter	Kürzel	Flugabschnitt		Einheit
		Steigflug	Reiseflug	
Bahnneigungswinkel	γ	5	0	$^\circ$
Radius des Flugkreises	r_{turn}	5.000	5.000	m
Fluggeschwindigkeit in EAS	v	9	9	m/s
Zielhöhe	h_{target}	20.000	-	m
Zielzeit	t_{target}	-	6	h
Art der Priorität	-	Höhe	Zeit	-

Tabelle 6.4: Bahndaten zur Mission_0

6.2.2 Analyse der Ergebnisse

Die Ergebnisse der Missionssimulation der Standardmission werden im Folgenden vorgestellt. Anhand dieser erfolgen Analysen und Schlussfolgerungen zur Stationierbarkeit. Zur Übersichtlichkeit werden nicht alle Parameter dargestellt, sondern nur die für den Missionserfolg relevanten Parameter bzw. diejenigen, die sich aus der Analyse ergeben. Es ist anzumerken, dass die Analyse lediglich anhand der Ausgaben der Simulation durchgeführt wird.

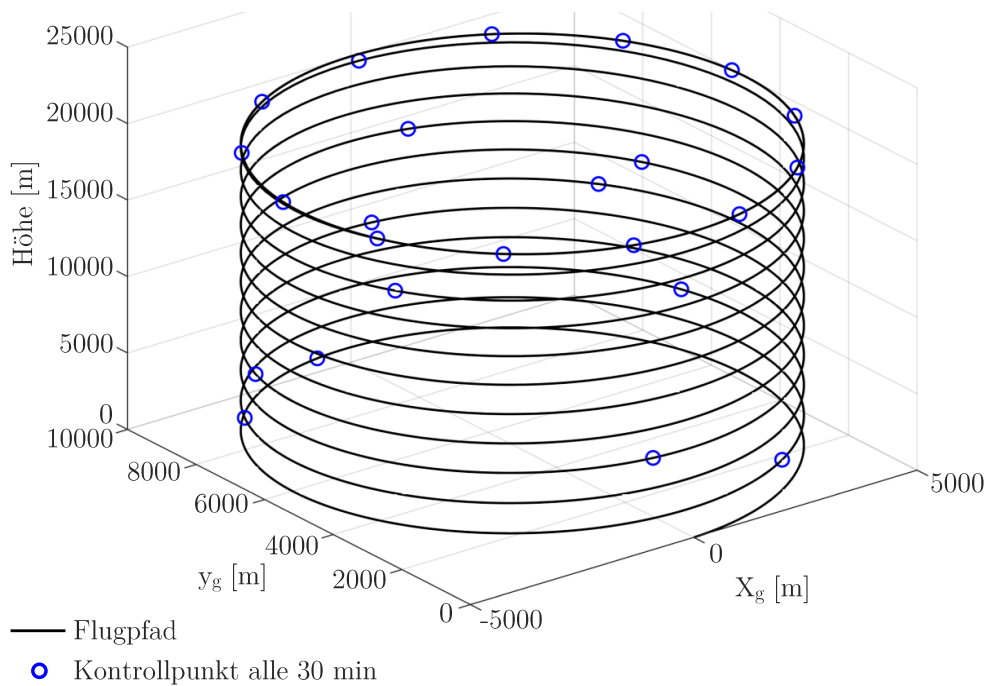


Abbildung 6.5: Flugpfad im geodätischen KOS

Das erste Ergebnis wird direkt an den Nutzer als Rückgabemeldung aus Matlab ausgegeben. Die Meldung **“end by: Mission accomplished”** bedeutet, dass die Mission komplett abgebildet wurde. Dies deckt sich mit der Untersuchung des Flugpfades in Abbildung 6.5. Darin ist sichtbar, dass die Flugabschnitte des Steig- und Reisefluges abgebildet wurden. Durch die Auswertung der Kontrollpunkte innerhalb der Abbildung kann ein Reiseflug in Zielhöhe von 6 h nachgewiesen werden. Alternativ ist der Verlauf der Mission anhand des Höhenverlaufs in Abbildung 6.6 nachvollziehbar. Darin wird sichtbar, dass die Steigrate, also die Vertikalgeschwindigkeit, mit wachsender Höhe zunimmt. Dies resultiert aus der Abnahme der Luftdichte und folglich der Zunahme der TAS.

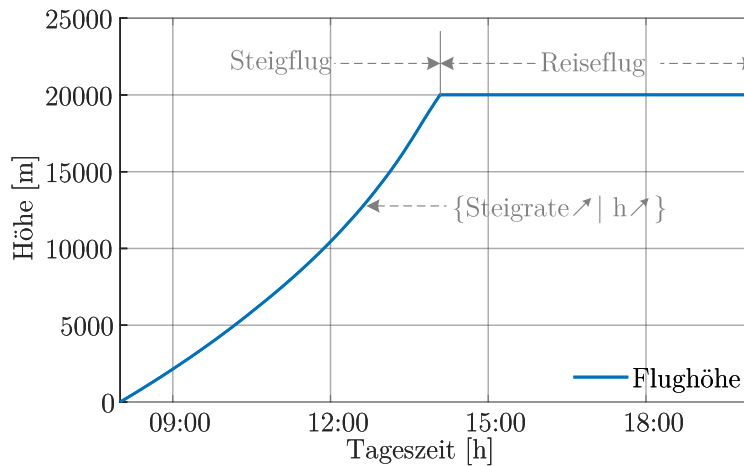


Abbildung 6.6: Verlauf der Höhe über die Missionszeit

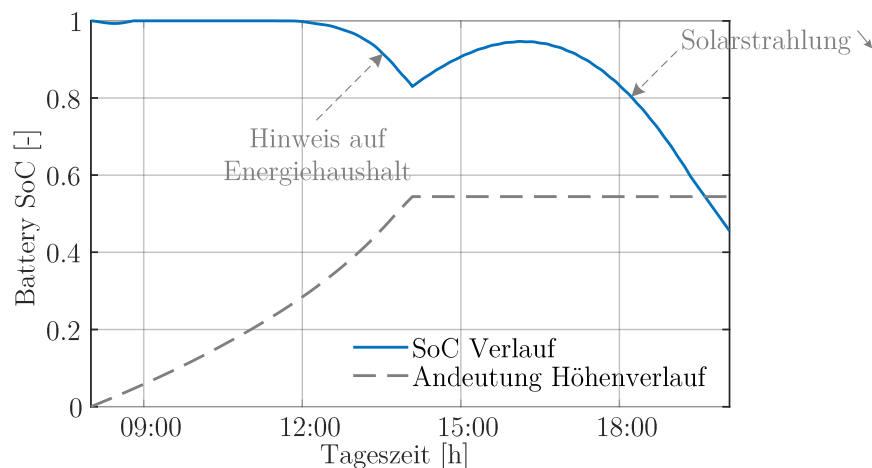


Abbildung 6.7: Verlauf des Batterieladestands über die Missionszeit

Der Verlauf der Batteriekapazität SoC ist in Abbildung 6.7 dargestellt und liegt zum Missionsende noch ca. 25 % über dem minimalen Ladezustand von 0,2. Zur besseren Visualisierung ist in der Abbildung der Höhenverlauf nicht maßstäblich eingeblendet. Daraus wird ersichtlich, dass zum Ende des Steigfluges der Energiehaushalt stärker belastet wird und die Batterie beginnt zu entladen. Im dahinter liegenden Reiseflug lädt die Batterie noch ca. 2 h. Der darauf folgende Abfall des SoC ist gekoppelt mit dem sinkenden Sonnenstand und die daraus resultierende Reduzierung der Solar Generatorleistung. Beide Hinweise werden zur weiteren Analyse genutzt.

Die Prüfung der gesamten elektrischen Leistung, dargestellt in Abbildung 6.8, ergibt, dass am Steigfluge die Leistungsgrenze des Motors erreicht wird. Diese liegt im vorliegenden Fall bei 2.000 W für einen Motor. Es ist anzumerken, dass neben der Leistung der Motoren noch die Systemversorgung und die Heizleistung der Batterien in den Gesamtverbrauch mit einfließen. Als Resultat des Leistungslimits ist der Motor genauer zu untersuchen. Des Weiteren ist in der Abbildung der Einfluss der Stellgröße γ_{max-da} vergrößert dargestellt. Die Reduzierung des Bahnneigungswinkels γ erzeugt einen sägezahnförmigen Leistungsabfall.

Der Verlauf der Solar Generatorleistung spiegelt die Modelle des Solarmodells wieder. Die sinusförmigen Wellen innerhalb des Hauptverlaufs in Abbildung 6.9 sind zurückzuführen auf die Kreisbahn des Flugzeuges. Die Variation der Periodendauer dieser Wellen kann durch den Anstieg der TAS erklärt werden. Des Weiteren ist ersichtlich, dass bereits bei Missionsbeginn solare Leistung verfügbar ist. Eine vorgezogene Startuhrzeit erhöht somit die Ausnutzung der max. vorhandenen solaren Strahlung. Folglich wird die Missionsstartzeit als Schlüsselparameter identifiziert.

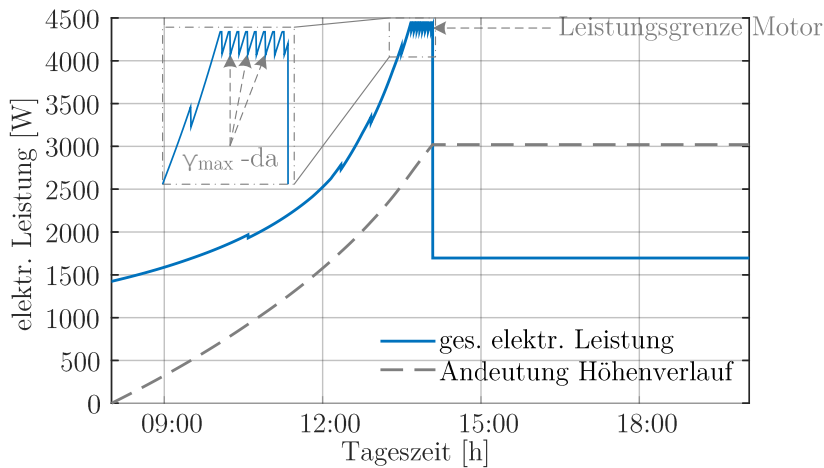


Abbildung 6.8: Verlauf der ges. elektr. Leistung über die Missionszeit

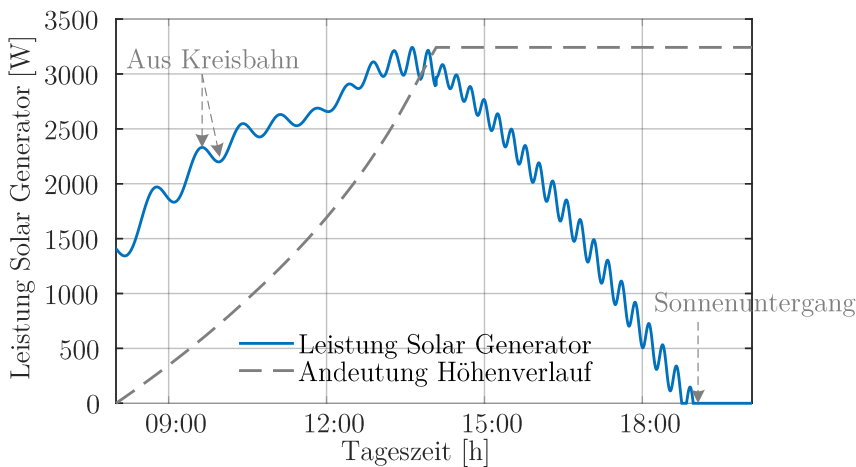


Abbildung 6.9: Verlauf der Solar Generator Leistung über die Missionszeit

Die weitere Untersuchung fokussiert auf den Motor. Dazu sind die Leistungsdaten eines Motors in Abbildung 6.10 dargestellt. Aus der Abbildung ist ersichtlich, dass der Motor unmittelbar nach Missionstart an seiner Drehmomentgrenze arbeitet. Weiterhin wird die Leistungsgrenze vor Ende des Steigfluges erreicht. In Abbildung 6.11 ist der Grenzbetrieb des Motors während des Steigfluges offensichtlich. Als Resultat werden das Drehmoment des Motors und die max. Motorleistung als missionsrelevante Parameter identifiziert.

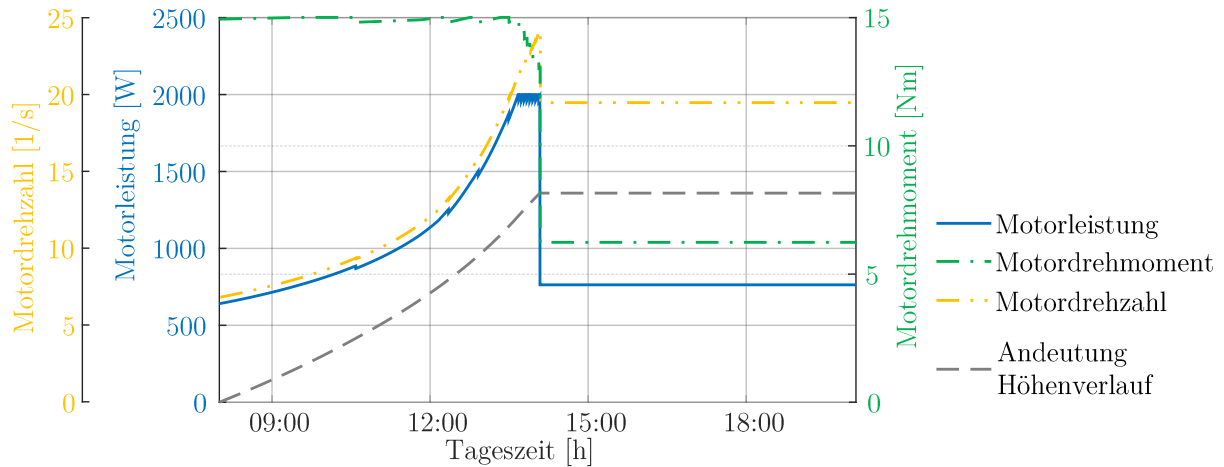


Abbildung 6.10: Leistungsparameter eines Motors über die Missionszeit

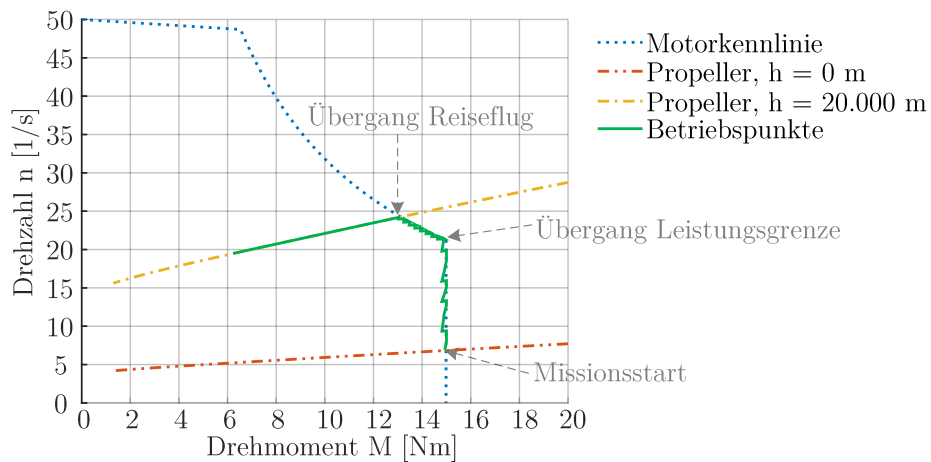


Abbildung 6.11: Verlauf der Betriebspunkte im Motorkennfeld

Zwischenfazit

Die Kopplung der Datenpunkte am Missionsende des SoC in Abbildung 6.7 und des Sonnenuntergangs in Abbildung 6.9 ergeben durch Extrapolation, dass die Batterien gegen ca. 22-23 Uhr den minimalen Ladestand erreichen und der Sinkflug eingeleitet werden muss. Daher ist eine durchgehende Stationierung über Nacht für die Standardmission nicht möglich. Jedoch kann die projektierte Reiseflughdauer von 6 h eingehalten werden.

Weitere Untersuchungen

Die Heizleistung der Batterien, dargestellt in Abbildung 6.12, weist ein Maximum zwischen 12 bis 13 Uhr auf. Dies entspricht einem Höhenbereich von 10.000 bis 11.000 m. Der Abfall der Heizleistung nach dem Maximum kann auf die konstante Temperatur bei stetig fallender Luftdichte in der Stratosphäre zurückgeführt werden. Dabei sinkt der konvektive Wärmegang, wodurch der Wärmeverlust sinkt.

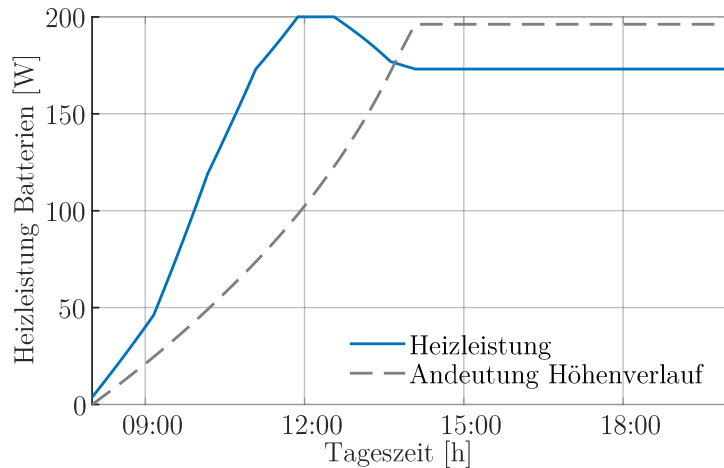


Abbildung 6.12: Heizleistung der Batterien über die Missionszeit

Der Verlauf der solaren Strahlung ϵ_{dir} in Abbildung 6.13 zeigt den Einfluss des Höhenwinkels α_S in Kombination mit steigender Flughöhe h . In 20.000 m Höhe während des Reisefluges ist die Wirkung des Höhenwinkels auf die solare Strahlung vernachlässigbar. Dies stimmt mit den Verläufen aus Abbildung 5.2 (siehe Kapitel 5.1.2) überein.

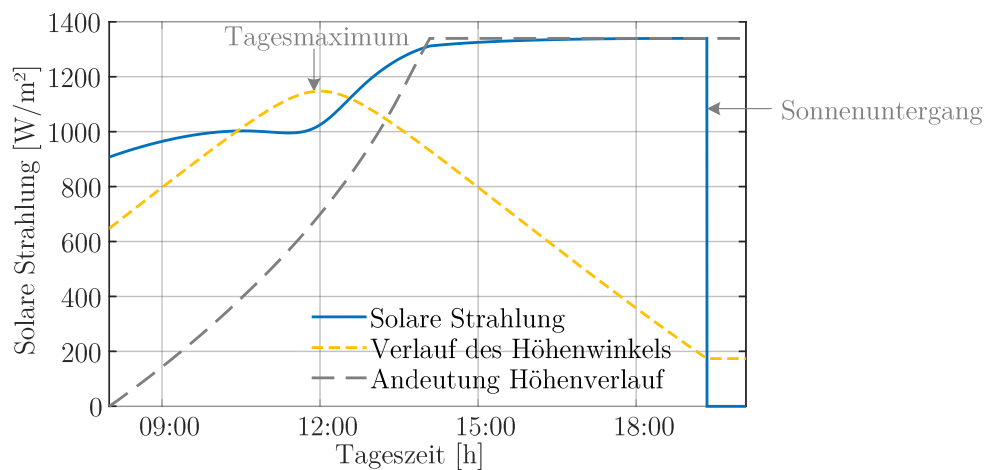


Abbildung 6.13: Solare Strahlung ϵ_{dir} über die Missionszeit

Der Bahnneigungswinkel γ wird infolge der Belastungsgrenze des Motors konstant reduziert. Der entsprechende Verlauf ist in Abbildung 6.14 dargestellt. Darin sind die Stufen durch die Reduzierung von γ ersichtlich. Der Wert von $3,6^\circ$ für den Bahnneigungswinkel kennzeichnet das Bahnwinkelmaximum des analysierten Flugzeuges.

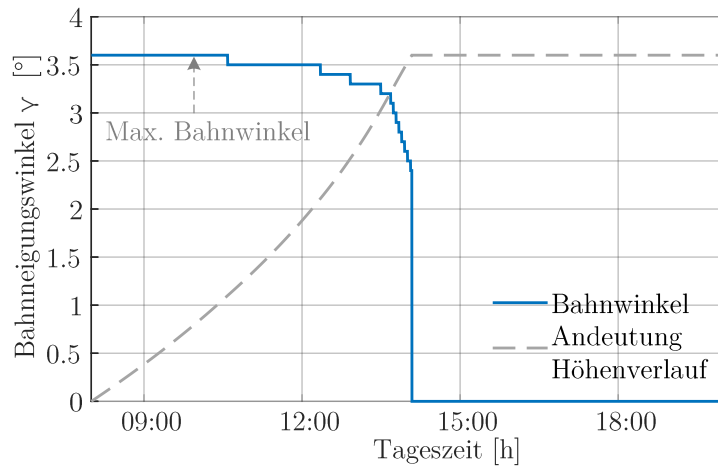


Abbildung 6.14: Bahnneigungswinkel γ über die Missionszeit

Der Verlauf des Auftriebskoeffizienten c_L ist durch die geringe Änderung des Bahnneigungswinkel γ annähernd konstant. Sein Wert beträgt im Mittel 0,758 bei einer Streuung von 0,02. Demgegenüber steigt der Widerstandskoeffizient mit der Flughöhe an, wie in Abbildung 6.15 ersichtlich. Dies resultiert aus der Verringerung der Reynoldszahl Re aufgrund der steigenden kinematischen Viskosität ν . Diese steigt im Vergleich zur TAS stärker an, welche ebenfalls die Reynoldszahl beeinflusst.

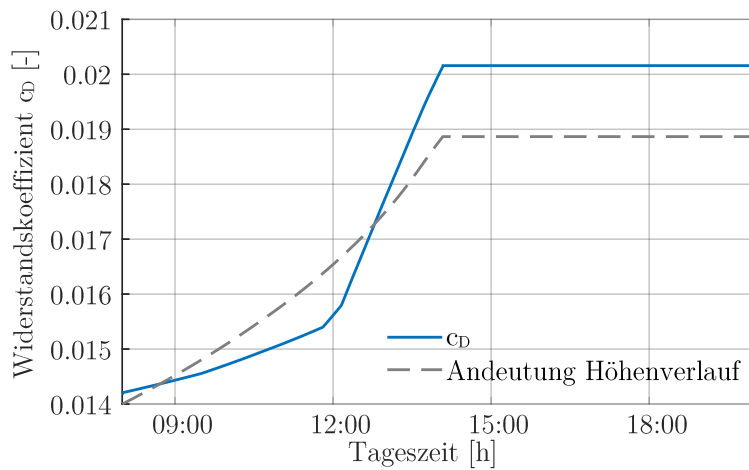


Abbildung 6.15: Widerstandskoeffizient c_D über die Missionszeit

6.2.3 Missionsoptimierung

Im vorangegangenen Abschnitt sind einige missionsrelevante Schlüsselparameter identifiziert worden, wovon einige im Folgenden variiert werden. Dabei wird ermittelt, wie die Parameter beeinflusst werden können, um die Mission, hinsichtlich des Batterieladestandes SoC am Missionsende, zu optimieren. Der entsprechende Wert des SoC für die Standardmission beträgt 0,464. Es werden die Parameter **Motordrehmoment** und die **Startzeit** der Mission untersucht. Es ist anzumerken, dass die Analyse zur Variation von Wirkungsgraden und Energiedichten sowie der Solarzellenfläche bewusst vernachlässigt wurde, da diese aufgrund des Komponentenstandes nach [1] für die Standardmission nicht geändert werden können.

Motordrehmoment

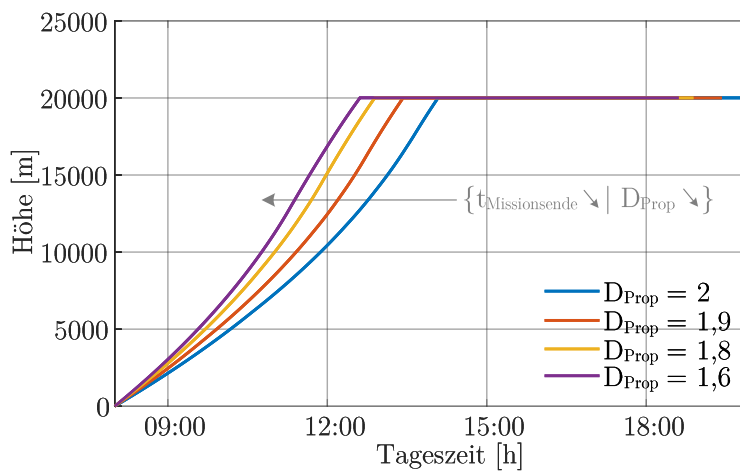


Abbildung 6.16: Höhenverlauf abhängig vom Propellerdurchmesser über die Missionszeit

Die Einflussfaktoren auf das Motordrehmoment können mit den Formeln 4.57, 4.59 und 4.60 aus Kapitel 4.2.7 ermittelt werden. Durch Einsetzen der Formeln ineinander ergibt sich Formel 6.1. Beeinflussbare Stellgrößen sind demnach der Leistungsparameter c_P , der Fortschrittgrad J , die Fluggeschwindigkeit v und der Propellerdurchmesser D_{Prop} . Da der Propellerdurchmesser kubisch in die Formel einfließt, wird dessen Einfluss genauer untersucht. Dabei werden sinkende Werte des Propellerdurchmessers untersucht.

$$M_{Prop} = \frac{c_P \cdot \rho}{J^2 \cdot 2 \cdot \pi} \cdot v^2 \cdot D_{Prop}^3 \quad (6.1)$$

mit: M_{Prop} : Drehmoment des Propellers [Nm]
 c_P : Leistungsparameter Propeller [-]
 ρ : Luftdichte [kg/m^3]
 J : Fortschrittgrad [-]
 v : Fluggeschwindigkeit [m/s]
 D_{Prop} : Propellerdurchmesser [m]

Die Ergebnisse der Parametervariation sind in Abbildung 6.16 dargestellt und zeigen, dass das Flugzeug mit kleineren Propellerdurchmessern steiler steigen kann und folglich die Reishöhe zeitlich früher

erreicht. Dies wirkt sich entsprechend auf den SoC aus wie Abbildung 6.17 zeigt. Entsprechende Verläufe des Drehmomentes und der Drehzahl sind jeweils in Abbildung 6.18 und 6.19 dargestellt. Dabei ist anzumerken, dass die Drehzahl bei sinkendem Propellerdurchmesser stark ansteigt.

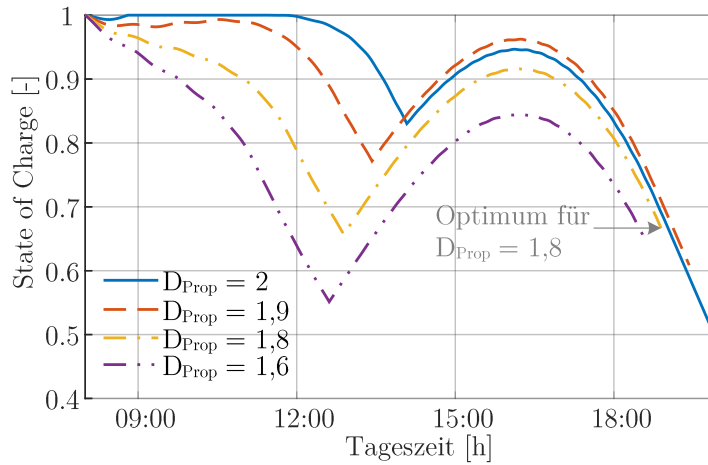


Abbildung 6.17: SoC abhängig vom Propellerdurchmesser über die Missionszeit

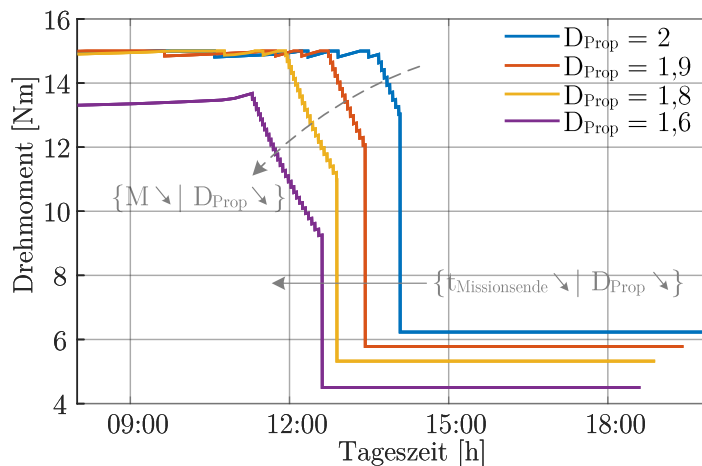


Abbildung 6.18: Motordrehmoment abhängig vom Propellerdurchmesser über die Missionszeit

Als Schlussfolgerung aus den Analysen wird ein Optimum des SoC für einen Propellerdurchmesser von 1,8 m erreicht. Der SoC beträgt am Missionsende 0,78. Jedoch ist anzumerken, dass dies nur für die geplante Mission eine Verbesserung bewirkt. Eine Simulation gleicher Gesamt-Missionsdauern ergibt für einen Propellerdurchmesser von 1,8 m ein SoC von 0,441, welcher niedriger ist als der Vergleichsfall mit 2 m Propellerdurchmesser. Dies resultiert aus einer Verschlechterung des Propellerwirkungsgrad η_{Prop} aufgrund der höheren Drehzahl. Die Drehzahl beeinflusst den Fortschrittgrad nach Formel 4.57, welcher wiederum in die Berechnung des Wirkungsgrades einfließt.

Aus dieser Analyse wird ersichtlich, dass die Simulationsergebnisse stets bewertet und validiert werden müssen um falsche Schlussfolgerungen zu vermeiden.

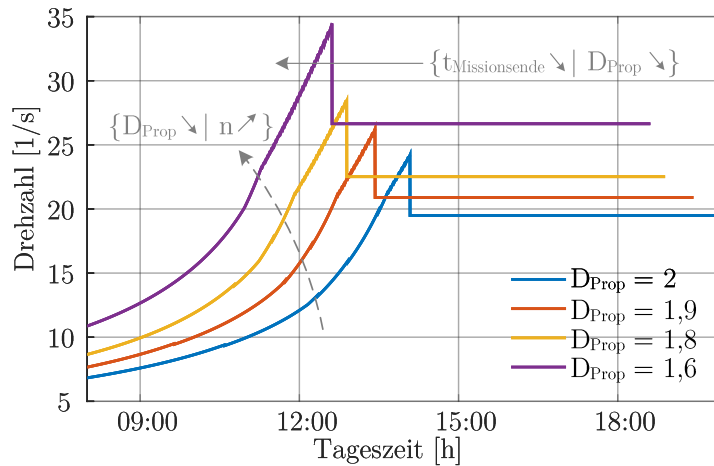
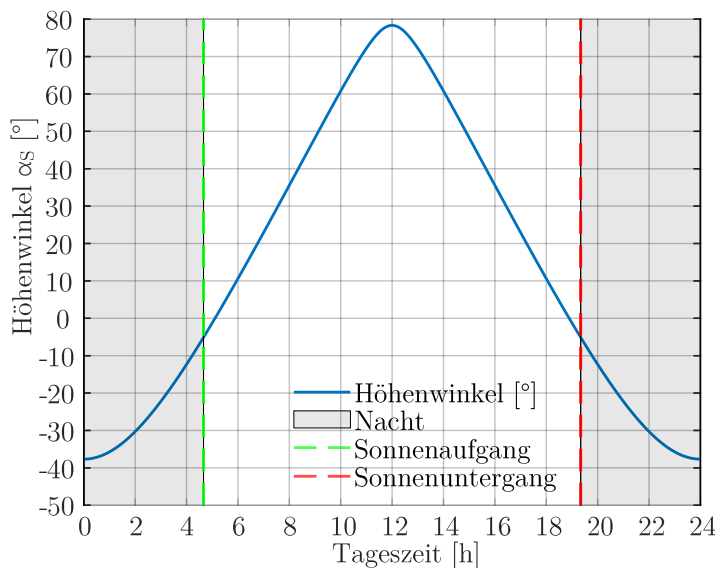


Abbildung 6.19: Motordrehzahl abhängig vom Propellerdurchmesser über die Missionszeit

Startuhrzeit

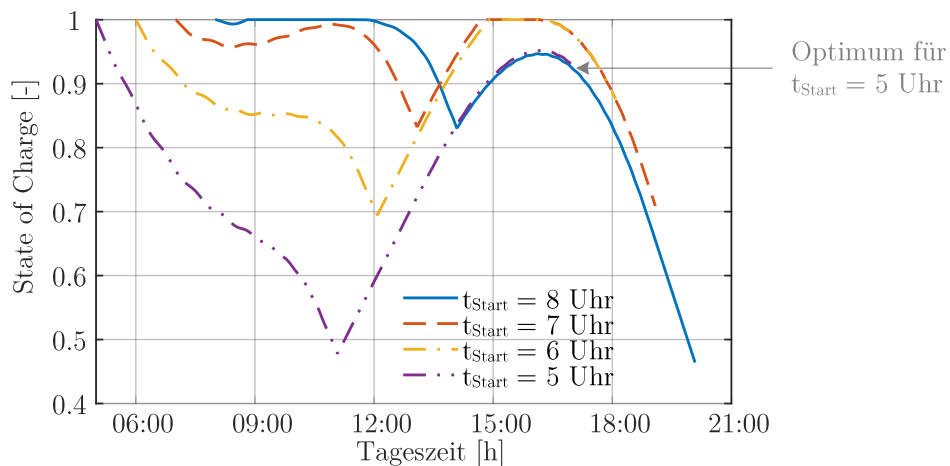
Zur Optimierung der Startuhrzeit wird zuvor der Verlauf des Höhenwinkel α_S für die gewählte Position und den Tag im Jahr analysiert. Der Sonnenstand ist nach Abbildung 6.20 um 12 Uhr am höchsten. Die Dauer der Standardmission beträgt knapp 12 h. Daher wird die Startuhrzeit t_{Start} um den Wert 6 Uhr variiert um theoretisch das Maximum des Integrals des Sonnenstandes zu erhalten.



Breitengrad $\phi = 32^\circ$ N
 Längengrad $\lambda = 0^\circ$
 Tag im Jahr $n = 142$

Abbildung 6.20: Höhenwinkel α_S für $\phi = 32^\circ$ N, $n = 142$, $h = 20.000$ m

Die Ergebnisse des SoC sind in Abbildung 6.21 dargestellt. Daraus ist ersichtlich, dass eine Startuhrzeit von 5 Uhr den höchsten Restladestand bei Missionsende ergibt. Der SoC am Missionsende beträgt 0,926, wodurch eine Verbesserung zur Standardmission von 46 % erzielt wird. Die zusätzliche Differenz des SoC zwischen t_{Start} 5 und 6 Uhr lässt sich auf den Anstieg der solaren Strahlung durch den Höhenzuwachs zurückführen. Dieser Zusammenhang kann in Abbildung 5.2 des Kapitel 5.1.2 nachgelesen werden. Die Untersuchung einer noch früheren Startuhrzeit führt zu niedrigeren SoC Werten.

Abbildung 6.21: Verlauf des SoC abhängig von der Startuhrzeit t_{Start}

Zwischenfazit

Die Missionsoptimierung erfordert ein breit gefächertes Fachwissen über die Funktionsweise der verschiedenen Flugzeugkomponenten. Des Weiteren sind Wechselwirkungen zwischen Parametern teilweise schwer vorhersehbar. Zum Zeitpunkt der Analysen lag der Programmstand vor, dass Parametervariationen manuell einzustellen sind und jede Simulation separat durchgeführt werden muss. Dies blockiert zeitliche Kapazitäten des Nutzers, weshalb der Bedarf einer Studienfunktion besteht.

6.3 Erweiterung für Parameterstudien

Im Zuge der Erweiterung der Simulation wird die Anwendung von Studien integriert. Das bedeutet, dass einzelne Parameter automatisiert variiert werden. Dabei wird für jede Variation eine neue Missionssimulation durchgeführt. Die Umsetzung erfolgt durch die zusätzliche Klasse **Study**. Die Einordnung in die Klassenstruktur, einzelne Funktionen sowie die Ausgabemöglichkeit werden im Folgenden beschrieben. Es ist anzumerken, dass keine Studienergebnisse im Detail aufgeführt werden.

6.3.1 Einschub Studienklasse und -funktionen

Die Struktur der Studie basiert auf dem Objekt **Study** der gleichnamigen Klasse. Unter ihr wird das Objekt Scenario erzeugt inklusive der weiteren Mid und Sub Level Klassen. Die so verwendete gesamte Klassenstruktur wird in Abbildung 6.22 dargestellt.

Die **Attribute** der Klasse und ihre **Funktionen** sind in Abbildung 6.23 aufgelistet. Bezüglich der Attribute findet sich neben dem bekannten Scenario das Feld **inputs**. Dieses enthält die Auswahldaten für das Aircraft, Mission und Environment zur Übergabe an das Scenario bei seiner Erstellung. **mode** beschreibt die Art der Studie. Es sind Studien möglich bei der entweder nur ein Parameter oder auch zwei Parameter gleichzeitig variiert werden. Das Attribut **param** enthält die für die Studie zu

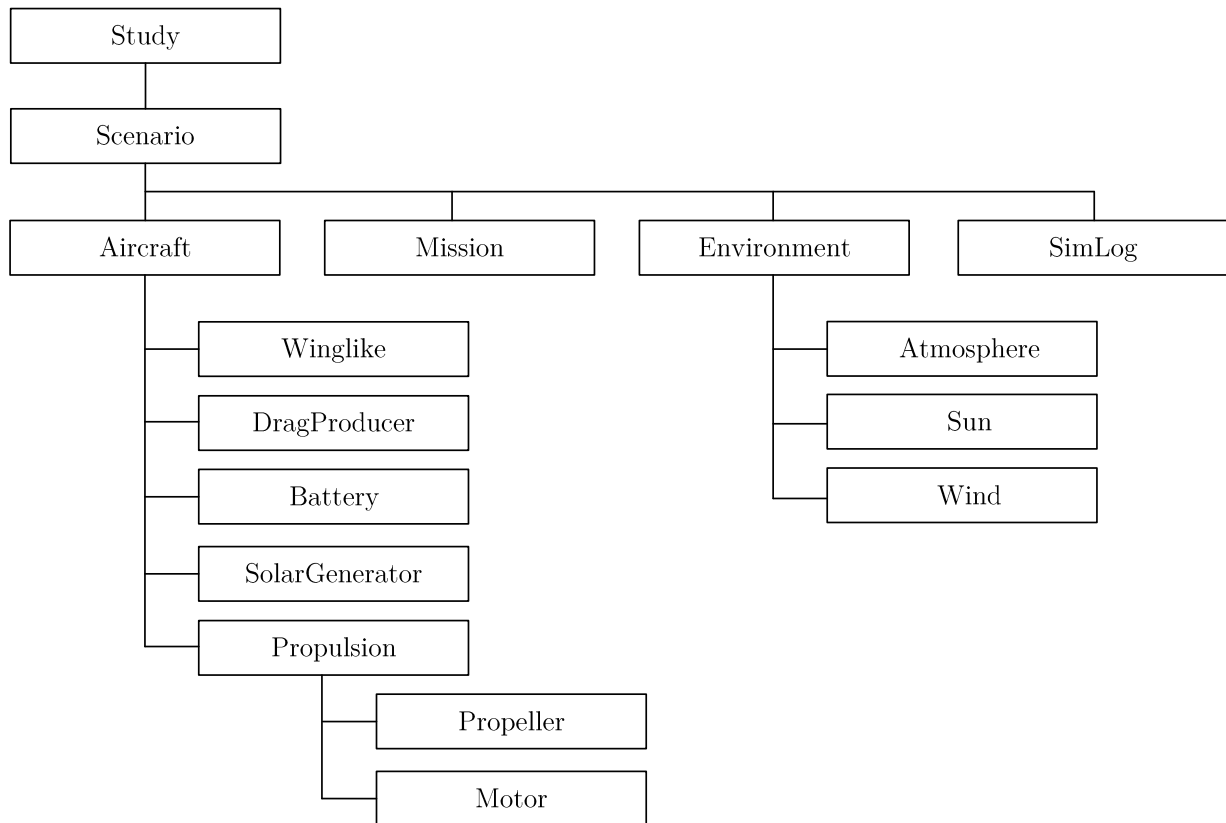


Abbildung 6.22: Gesamte Klassenstruktur für Studien

variierenden Parameter. In **paths** sind die entsprechenden Pfade der Parameter hinterlegt, um diese innerhalb des Szenarios ändern zu können. **progress**, **state** und **timer** dienen der internen Ermittlung des Studienfortschrittes zur aktiven Rückmeldung an den User während der Studie. **StudyLog** enthält die gespeicherten Daten aus jeder Einzelsimulation eines Szenarios. Hier werden nicht alle Daten gespeichert, sondern nur die vom Benutzer ausgewählten. Beispielsweise wird der min. SoC der Batterien eines Studiendurchlaufes abgespeichert.

Study	<u>Attributbezeichnung</u>	<u>Einheit</u>
Scenario	Objekt	-
index	Interner Zähler	[-]
inputs	Feld mit Eingabeauswahl	-
max_rounds	Anzahl der Studienrunden	[-]
mode	Art der Studie	-
param	Feld mit Parameterdaten für Studie	-
paths	Feld mit Pfadangaben der Studienparameter	-
progress	Interne Studienstandsanzeige	[-]
state	Status der Studie	-
StudyLog	Feld mit gesicherten Simulationsdaten	-
timer	Feld für Zeiten zu speichern	[s]
LogData		
SetData		
Studying		
PlotData		
PushStudy		

Abbildung 6.23: Klasse Study inklusive Bezeichnungen

Die Funktionen erklären sich anhand des gesamten Studienablaufs nach Abbildung 6.24. Im Auslöseskript PerformSimulation wird unterschieden, ob eine Studie aktiv ist oder nicht. Die Alternative einer einfachen Simulation besteht weiterhin. Für den Fall in Abbildung 6.24 ist die Studie aktiv. Dabei wird zuerst das Objekt Study erzeugt und darauf folgend die Funktion **Study.studying** ausgeführt. Diese erzeugt beginnend ein Scenario, welches wiederum automatisch sämtliche untergeordneten Objekte instanziiert. Die Funktion **Study.SetData** greift auf die für die Studie relevanten Parameter innerhalb des Scenarios zu und überschreibt diese mit den für die Studie geforderten. Danach erfolgt die bekannte Funktion Scenario.simulate, die eine komplette Missionssimulation durchläuft. Nachdem diese beendet ist, speichert die Funktion **Study.LogData** die durch den Benutzer vorgegebenen Daten aus dem Scenario heraus. Das Scenario wird anschließend gelöscht und der interne Index wird erhöht. Die Funktion **Study.PushStudy** setzt die neuen Studienparameter des nächsten Durchlaufs. Die Schleife setzt dann wieder am Punkt der Erstellung des Scenarios an bis alle Studienparameter durchlaufen sind.

Eine Studie kann je nach Schrittweite der zu variierenden Studienparameter und Simulationsumfang signifikante Laufzeiten annehmen. Im Zuge der Erarbeitung wurden Zeiten größer 60 min erreicht trotz geringer Schrittweite und Parametervariation. Aufgrund dessen wird dem Benutzer nach jedem Studiendurchlauf eine Angabe zum aktuellen Studienfortschritt in Prozent und die Angabe der restlichen Studienzeit gegeben. Letztere wird pro gesamtem Simulationsdurchlauf neu berechnet auf Basis der noch durchzuführenden Simulationen und der bei der letzten Simulation aufgewendeten Rechenzeit. Es ist wichtig die aktuell letzte Durchlaufzeit zu berücksichtigen, da je nach Arbeitsspeicher und parallelen Anwendungen des Nutzers die Gesamtzeit um Stunden variieren kann. Folglich ändert sich die Angabe der Restzeit dynamisch.

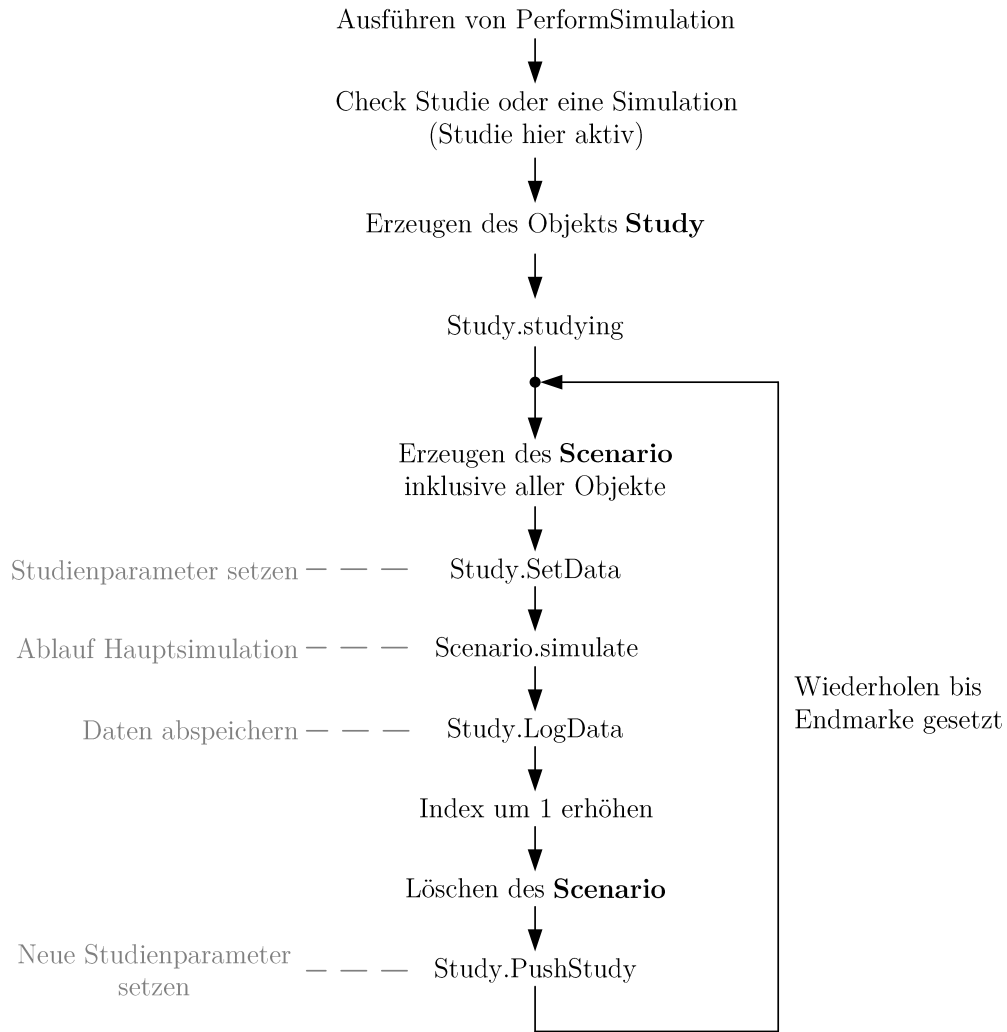


Abbildung 6.24: Ablauf der Studie

6.3.2 Potential Studien

Mit den vorgestellten Methoden können verschiedene Arten von Studien durchgeführt werden. Einfache Studien mit der Variation eines Parameters sind sinnvoll für komponentenbezogene Parameter wie beispielsweise in Tabelle 6.5 angegeben. Darin sind ebenfalls mögliche Analyseziele gezeigt.

Studienparameter	Ziel der Analyse
Flugzeugmasse	Flugleistungen
Spannweite des Flügels	Flugleistungen
Energiedichte der Batterien	Stationierbarkeit
Wirkungsgrad Solarzellen	Batteriegewicht
Propellerkennfeld	Steigleistung

Tabelle 6.5: Mögliche Ziele für Ein-Parameter-Studien

Die Variation von zwei Parametern ist für Standort- und zeitbezogene Studien sinnvoll. Diese zielen meist nur auf die Stationierbarkeit ab. Beispiele dazu sind in Tabelle 6.6 gegeben. Beliebige Kombinationen der gezeigten Parameter sind dennoch möglich. Für fortgeschrittene Analysen können Zwei-Parameter-Studien für Komponenten-Kombinationen durchgeführt werden.

Studienparameter	Ziel der Analyse
Längen- & Breitengrad	Stationierbarkeit
Tag im Jahr & Tageszeit	Stationierbarkeit
Breitengrad & Tag im Jahr	Stationierbarkeit
beliebige Kombinationen	abhängig nach Nutzer
...	...

Tabelle 6.6: Mögliche Ziele für Zwei-Parameter-Studien

6.3.3 Beispielstudien

Heiztemperatur der Batterien

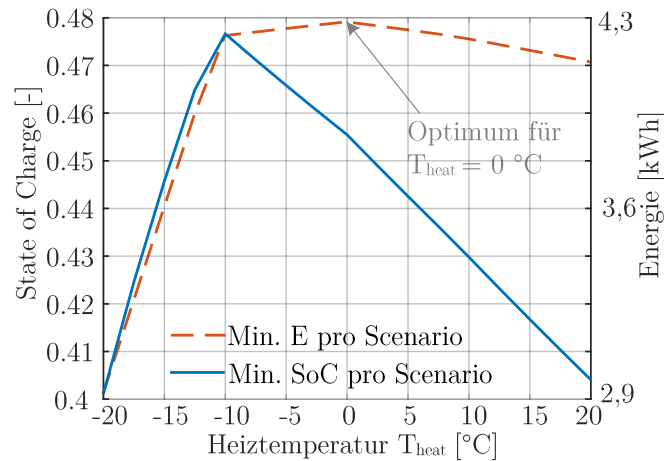
Im folgenden Beispiel wird die Heiztemperatur der Batterien variiert zum Auffinden eines sogenannten "Sweet-Spots". Dieser entspricht dem Optimum eines Parameters zur Maximierung der von diesem Parameter abhängigen Missionseffizienz ausgedrückt durch den SoC. Die restlichen Parameter der Studie entsprechen denen der Standardmission. Der untersuchte Wertebereich des Studienparameters kann aus Tabelle 6.7 entnommen werden.

Der optimale SoC ergibt sich für eine Heiztemperatur von $-10\text{ }^{\circ}\text{C}$, dargestellt in Abbildung 6.25. Der eckig-förmige Verlauf ergibt sich in diesem Fall aufgrund der Genauigkeit des hinterlegten Kennfeldes für die Batterieheizung. In anderen Fällen kann die Schrittweite des Studienparameters einen ähnlich förmigen Verlauf erzeugen. Daher sind kleine Werte für die Schrittweite zu empfehlen, die jedoch die Gesamtdauer der Studie erhöhen. Des Weiteren ist in Abbildung 6.25 die Restenergie des jeweiligen Simulationsdurchlaufes aufgeführt. Die max. Kapazität der Batterie wird durch die Heiztemperatur beeinflusst, wodurch je nach Temperatur der Batterien die max. verfügbare Energie stagniert.

Bezeichnung	Wert	Einheit
Parametername	Heiztemperatur Batterien	
Parameterkürzel	T_{heat}	
min. Studienwert	-20	°C
max. Studienwert	20	°C
Schrittweite	1	°C

Tabelle 6.7: Wertebereich des Studienparameters T_{heat}

Folglich liegt das Optimum für die Gesamtenergie bei 0 °C. Es ist anzumerken, dass dieses Ergebnis nicht pauschal für alle Missionen übernommen werden kann, da der Wert für die Heizleistung der Batterien höhenabhängig ist. Abweichende Zielhöhen zur Standardmission erfordern eine erneute Betrachtung.

Abbildung 6.25: Min. SoC und elektr. Energie E der Studie über Heiztemperatur T_{heat}

Starttag der Mission

Die Studie bzgl. des Missionstages hat das Ziel den Tag im Jahr mit dem höchsten SoC bei Missionsende zu ermitteln. Der Wertebereich des zu variierenden Parameters n ist in Tabelle 6.8 gegeben. Dabei wird das ganze Jahr betrachtet mit einer Schrittweite von 5 Tagen.

Bezeichnung	Wert	Einheit
Parametername	Tag im Jahr	
Parameterkürzel	n	
min. Studienwert	0	—
max. Studienwert	365	—
Schrittweite	5	—

Tabelle 6.8: Wertebereich des Studienparameters n

Die Ergebnisse der Studie sind in Abbildung 6.26 gezeigt und weisen den höchsten SoC bei Missions-

ende zwischen Tag 170 und 175 auf. Folglich sollte die Mission, für beispielsweise eine Flugkampagne, im Bereich um Tag 175 zeitlich gewählt werden. Der starke Abfall des SoC an den Rändern in Abbildung 6.26 resultiert aus der Funktionsweise des Programms. Liegt der SoC innerhalb einer Mission unterhalb des min. SoC wird dieser zu null gesetzt.

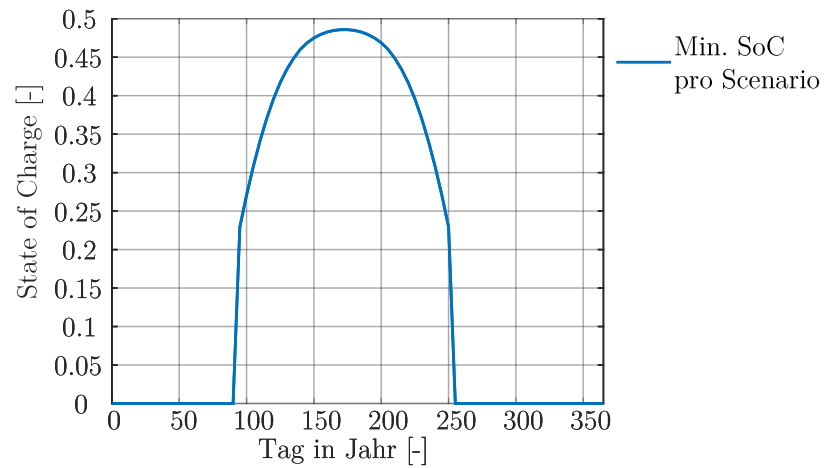


Abbildung 6.26: Min. SoC der Studie über Tag im Jahr n

7 Zusammenfassung und Ausblick

7.1 Fazit

Ziel dieser Abschlussarbeit war die Entwicklung einer Missionssimulation für das solarelektrisch betriebene Statrosphärenflugzeug HAP. Die Umsetzung erfolgte durch die Software Matlab mithilfe der objekt-orientierten Programmierung. Dadurch ergab sich die Möglichkeit die gesamte Programmstruktur auf Basis des realen Flugzeuges aufzubauen. Die so entstandenen Programmgruppierungen wurden dargelegt und erlauben eine intuitive Navigation innerhalb der Programmklassen.

Die Modellierung des Flugzeuges erfolgte auf Basis bestehender Literatur und entsprechender Formeln. An mehreren Positionen wurde auf die Nutzung vorhandener Kennfelder zurückgegriffen, die eine höhere Aussagekraft der berechneten Werte ermöglichen sollen. Die durchzuführende Mission kann in beliebige Abschnitte segmentiert werden. Dies erlaubt auch Untersuchungen verschiedener Flugstrategien zum Erreichen des geplanten Zieles, meist der Zielhöhe oder die Aussage zur Stationierbarkeit. Die Daten der Umgebung basieren primär auf vorhandenen Datensätzen und geometrischen Koordinatentransformationen. Anzumerken ist, dass der Einfluss des Windes mitverarbeitet wird, jedoch in der Pfadplanung der Mission nicht berücksichtigt ist.

Die Änderung von Eingabedaten erfolgt zentral innerhalb einer gesonderten Ordnerstruktur zur Funktionalität. Eine einzelne Datei konnte nicht realisiert werden, da mehrere Kennfelder als Eingabedaten eingelesen werden. Weiterhin wird durch die jeweiligen Datenfiles der Bezug zu den Objekten verdeutlicht und erleichtert wiederum das Programmhandling.

Die Verifikation der Simulation mithilfe weiterer Flugzeugkonfigurationen konnte nicht aussagekräftig durchgeführt werden. Dies ist zu begründen durch fehlende und nicht hinreichend genaue Daten zu den Flugzeugen und/oder ihrer jeweils durchgeführten Missionen. Jedoch wurden nicht-formale Methoden der Informatik vorgestellt um Abschnitte und verwendete Modelle der Simulation zu verifizieren. Des Weiteren konnte das Solarmodell anhand bestehender Literatur bestätigt werden.

Zuletzt wurden die Simulationsergebnisse zum vorhandenen Projektstand HAP vorgestellt. Der Einfluss numerischer Methoden und Stellgrößen auf die Simulation wurde dargelegt. Darauf aufbauend wurde die Standardmission analysiert und entsprechende Missionsoptimierungen untersucht. Im Zuge der Analyse entstand auch die Erweiterung der Simulation zur Durchführung von Parameterstudien. Diese sind mit bis zu zwei Parametern gleichzeitig durchführbar.

Abschließend ist zu betonen, dass die entwickelte Missionssimulation das Flugzeug HAP zwar erfolgreich abbilden kann, jedoch dient das Programm vor allem als Ausgangsbasis für Erweiterungen. Das Konstrukt der Simulation ist vergleichbar mit den sich verzweigenden Ästen eines Baumes. Je weiter die Simulation wächst desto tiefer wird die Detaillierung und Verfeinerung der Programmstrukturen. Dennoch ist die stetige Verfeinerung von Berechnungen und weitere Aufschlüsselung von Komponenten zukünftig anzustreben.

7.2 Ausblick

Die vorgelegte Programmstruktur ergibt umfängliche Ansatzpunkte für weitere Forschungsarbeiten. Dadurch offeriert das Simulationsprogramm durch seinen intuitiv gestalteten Aufbau und die Nutzung der Objektorientierung die Möglichkeit Änderungen und Erweiterungen einfach und schnell zu integrieren. Einige Anstöße zur Fortsetzung der Arbeit werden im Folgenden beschrieben. Es werden jeweils die betreffenden Programmklassen genannt.

Die Mission kann durch eine Pfadvorgabe nach Wegpunkten ergänzt werden. Dieser Ansatz verfolgt die Abbildung gesonderter Flugstrecken direkt nach dem Start bzw. kurz vor der Landung aufgrund beschränkter Lufträume oder Sperrgebiete. Außerdem könnte ein frei wählbarer Flugpfad durch die jeweilige Nutzlast oder Mission notwendig werden. Zu den Wegpunkten muss die Art des Anfluges eingebunden werden. Damit ist gemeint, wie das Flugzeug den nächst gelegenen Wegpunkt anfliegen soll. Es wird unterteilt in einen direkten Anflug oder auf einer Kreisbahn. Kombinationen sind ebenfalls wünschenswert. Theoretisch sind hier auch klassische Anschnittverfahren aus der Funknavigation denkbar oder die Untersuchung eines energetisch sinnvollen Anfluges. Neben dieser Integration kann die automatische Bahnkorrektur bei Pfadabweichung, beispielsweise durch Windeinfluss, eingearbeitet werden.

Der schiebende Flugzustand und die daraus resultierenden Leistungseinbußen bilden die konsequente Erweiterung nach der Integration des Windeinflusses. Dazu gehört auch die Untersuchung des schiebenden Propellers im Antriebsstrang.

Durch Missionsanalysen sind detaillierte Untersuchungen bezüglich der Rekuperation der Motoren durchzuführen. Dies könnte durch gezielten Höhenabbau während der Nachtphasen zu erweiterten Stationierungszeiten führen. Dieser Ansatz wird bereits bei anderen Betreibern von Solarflugzeugen verfolgt.

Das Solarmodell ist in Hinblick auf diffuse Strahlung insbesondere während der Start- und Landephase sowie im Steigflug zu erweitern. Die diffuse Strahlung wird erzeugt durch gestreutes Sonnenlicht an Wolken oder Dunst. Da sich diese Wetterphänomene meist in der unteren Troposphäre ereignen, sind sie vor allem in den zuvor genannten Flugphasen zu untersuchen.

Die Überwachung von Parametern innerhalb der Simulation kann durch zwei Methoden erweitert werden. Eine Methode besteht in der Kontrolle von Wertebereichen individueller Parameter zur Einhaltung plausibler Berechnungen. Dabei könnte die Simulation weiter simulieren, sofern die Grenzkontaktierung deutlich während der Simulationsdurchführung und in den Ergebnissen durch den Benutzer erkannt werden kann. Ein anderer Ansatz verfolgt die dynamische Entwicklung von Werten. Dabei soll das Differential eines Parameters überwacht werden um einen zu hohen und eventuell falschen Wertezuwachs abzufangen.

Im Zuge der Erweiterung der Simulation wird die benötigte Zeit zur Berechnung der Simulation stetig größer. Daher sollte untersucht werden, wie die Simulationszeit reduziert werden kann. Ein Ansatz besteht durch die optionale Abwahl der Kontrollfunktionen, beispielsweise die Funktion zum Überprüfen der Flugbereichsgrenzen. Alternativ kann die Klasse Mission erweitert werden zur intelligenteren Vorgabe der Bahndaten. Diese können durch ein entsprechendes Kennfeld, welches die max. Flugleistungen über der Höhe enthält, angegeben werden um Iterationsdurchläufe zu eliminieren.

Mit der Studienklasse sind Parameterstudien empfehlenswert. Komponentenweise Studien sind aufgrund des aktuell vorliegenden Projektverlaufs zunächst zu fokussieren um eventuell signifikante Verbesserungen noch vor Baubeginn implementieren zu können. Darauf folgend sind Standort- und Startterminstudien sinnvoll. Die Wahl des geeigneten Starttermins kann auf Basis der bisherigen Untersuchungen missionsentscheidend sein.

Der Einfluss des Thermalmodells der Batterie ist durch eine Sensitivitätsstudie zu bestimmen. Dabei sind die in dieser Arbeit verwendeten gemittelten Werte für die Energiedichtenskalisierung für separate Batterieblöcke anzulegen und zu integrieren. Des Weiteren ist die Degradierung abhängig von der Ladezyklenanzahl genauer zu untersuchen und eventuell ein entsprechendes Batteriemodell zu entwickeln.

Passend zur Studienklasse ist die Implementierung eines Graphical User Interface sinnvoll. Die erleichterte Bedienung des Simulationsprogrammes speziell bei der Auswahl und Einstellung der Studienparameter kann wertvolle Rechenkapazitäten sichern. Daneben kann eine alternierende Darstellung der Studienergebnisse die Erfolgsquote der Analysen fördern.

Weitere Analysen sind in Bezug auf den Erstflug des realen Flugzeuges zu erarbeiten um einen Abgleich zu den Simulationsdaten herzustellen. Ein geeigneter Versuchsplan inklusive der benötigten Flugzeug- und Missionsdaten ist zu erstellen. Mithilfe dieser soll die Vorhersagegenauigkeit der Simulation verifiziert werden. Neben dieser können Abweichungen von Simulation zu realem Flug reduziert werden um die Simulation zu verbessern. Als einfaches Beispiel ist der Horizontalflug zu nennen. Die resultierenden Daten aus erforderlicher Leistung und Fluggeschwindigkeit könnten einen Kalibrierfaktor für die Widerstandspolare ergeben. Darüber hinaus sind bereits bei Bodentestläufen wichtige Werte verfügbar wie beispielsweise Batteriekapazitäten oder Be- und Entladeverluste im Verbund des Gesamtsystems. Ein kontinuierlicher Abgleich der Simulationsparameter mit frischen Daten ist zielführend für genauere Ergebnisse.

Literatur

- [1] interne Quellen des DLR; DLR Braunschweig; letzter Zugriff am 20.12.2021; Seite: 1, 3, 6, 36, 37, 38, 42, 43, 44, 45, 48, 49, 57, 59, 65, 75, IV, V, X, XIV
- [2] High Altitude Long Endurance; Wikipedia; Zugriff am 09.10.2021; https://de.wikipedia.org/wiki/High_Altitude_Long_Endurance; Seite: 3
- [3] BAE fliegt PHASA-35; Flug Revue; Zugriff am 12.10.2021; <https://www.flugrevue.de/militaer/ unbemanntes-solarflugzeug-bae-fliegt-phas-a-35/>; Seite: 4, VIII, X
- [4] BAE Systems PHASA-35; Wikipedia; Zugriff am 12.10.2021; https://en.wikipedia.org/wiki/BAE_Systems_PHASA-35; Seite: 4, VIII, X
- [5] PHASA-35 completes critical endurance trials; AEROSPACE Manufacturing; Zugriff am 12.10.2021; <https://www.aero-mag.com/phas-a-35-completes-critical-endurance-trials/>; Seite: 3, VIII, X
- [6] Solarflugzeug; Wikipedia; Zugriff am 12.10.2021; <https://de.wikipedia.org/wiki/Solarflugzeug#Phasa-35>; Seite: 4, VIII
- [7] PHASA-35 Solar-Powered HALE UAV; AIRFORCE Technology; Zugriff am 14.10.2021; <https://www.airforce-technology.com/projects/phas-a-35-solar-powered-hale-uav/>; Seite: VIII
- [8] Zephyr Pioneering the Stratosphere; AIRBUS; Zugriff am 18.10.2021; <https://www.airbus.com/defence/uav/zephyr.html>; Seite: 4, IX, X, XI
- [9] Zephyr 2021 Test Flight Campaign - Take off; AIRBUS; Zugriff am 18.10.2021; https://www.airbus.com/defence/uav/zephyr.html#medialist_321921732-image-image-all_ml_1-2; Seite: 4
- [10] Zephyr bleibt 18 Tage in der Stratosphäre; Flug Revue; Zugriff am 16.10.2021; <https://www.flugrevue.de/militaer/airbus-solar-high-altitude-platform-system-zephyr-bleibt-18-tage-in-der-stratosphaere/>; Seite: 4, IX, X, XI
- [11] Airbus Zephyr; Wikipedia; Zugriff am 16.10.2021; https://de.wikipedia.org/wiki/Airbus_Zephyr; Seite: 4, IX
- [12] Zephyr; AIRBUS; Zugriff am 16.10.2021; https://www.airbus.com/defence/uav/zephyr.html#medialist-image-infographic-all_ml_2-3; Seite: IX
- [13] e-Genius; Wikipedia; Zugriff am 01.09.21; <https://de.wikipedia.org/wiki/E-Genius>; Seite: 4, X, XI
- [14] Alexander Schleicher präsentiert die AS 33; Alexander Schleicher Segelflugzeugbau; Zugriff am 01.09.21; <https://www.alexander-schleicher.de/alexander-schleicher-praesentiert-die-as33/>; Seite: 4, X, XI
- [15] Elektrisch eigenstartfähig: Die AS 33 Me; Aerokurier; Zugriff am 01.09.21; <https://www.aerokurier.de/segelflug/alexander-schleicher-elektrisch-eigenstartfaehig-as-33-me/>; Seite: 4, X, XI
- [16] Icaré II; Wikipedia; Zugriff am 01.10.21; https://de.wikipedia.org/wiki/Icar%C3%A9_II; Seite: 4, 5, X, XI

- [17] Andreas Bierig: *Design Considerations for High Altitude Platforms* (DLRK Vortrag, 01.09.2021); Seite: 6, 7, 8
- [18] Meilensteine für die Batterie- und Energieforschung; analytica; Zugriff am 18.01.22; <https://analytica.de/de/presse/trendberichte/batterieforschung/>; Seite: 6
- [19] Andreas Strohmayer; Universität Stuttgart: *Script: Flugzeugentwurf I*; Rev. 1.03; Seite: 9
- [20] Pacelab APD; Pace; Zugriff am 03.12.21; <https://pace.txtgroup.com/products/preliminary-design/pacelab-apd/>; Seite: 9
- [21] Flugzeugvorentwurfstool Pacelab APD in neuem Release; PRESSEBOX; Zugriff am 02.12.21; <https://www.pressebox.de/pressemitteilung/pace-aerospace-engineering-and-information-technology-gmbh/Flugzeugvorentwurfstool-Pacelab-APD-in-neuem-Release/boxid/278273>; Seite: 9
- [22] Pacelab Mission Suite; Pace; Zugriff am 03.12.21; <https://pace.txtgroup.com/products/aircraft-analysis-evaluation/pacelab-mission-suite/?hsLang=en>; Seite: 9, 10
- [23] Aerospace Blockset; Mathworks; Zugriff am 10.12.21; <https://de.mathworks.com/products/aerospace-blockset.html>; Seite: 10
- [24] Umgebungsmodelle; Mathworks; Zugriff am 10.12.21; <https://de.mathworks.com/products/aerospace-blockset.html#environment-models>; Seite: 10
- [25] Electrical Component Analysis for Hybrid and Electric Aircraft; Mathworks; Zugriff am 19.12.21; <https://de.mathworks.com/help/aeroblks/Electrical-Component-Analysis-Hybrid-and-Electric-Aircraft.html>; Seite: 10
- [26] Programmierparadigma; Wikipedia; Zugriff am 28.11.2021; <https://de.wikipedia.org/wiki/Programmierparadigma>; Seite: 11
- [27] Objektorientierte Programmierung; Wikipedia; Zugriff am 28.11.2021; https://de.wikipedia.org/wiki/Objektorientierte_Programmierung; Seite: 11
- [28] Ulrich Stein: *Objektorientierte Programmierung mit MATLAB* (Carl Hanser Verlag München, 2016); Seite: 11
- [29] UML Klassendiagramm – So behältst du bei der Objektorientierung den Überblick!; Codeadventurer; Zugriff am 29.11.2021; <http://www.codeadventurer.de/?p=2428>; Seite: 11
- [30] U.S. Standard Atmosphere, 1976; NASA; Zugriff am 10.06.21; https://www.ngdc.noaa.gov/stp/space-weather/online-publications/miscellaneous/us-standard-atmosphere-1976/us-standard-atmosphere_st76-1562_noaa.pdf; Seite: 29
- [31] Standard Atmosphere Functions; Mathworks; Zugriff am 10.06.21; <https://de.mathworks.com/matlabcentral/fileexchange/28135-standard-atmosphere-functions>; Seite: 29
- [32] Normatmosphäre; Wikipedia; Zugriff am 14.11.21; <https://de.wikipedia.org/wiki/Normatmosph%C3%A4re>; Seite: 29, 30
- [33] Erdradius; Wikipedia; Zugriff am 12.06.21; <https://de.wikipedia.org/wiki/Erdradius>; Seite: 30
- [34] Bernhard Keidel: *Auslegung und Simulation von hochfliegenden, dauerhaft stationierbaren Solardrohnen* (München, Universität München, 2000); Seite: 30, 31, 32, 61, 62, V

- [35] Daniel Ackermann *Echtzeitsimulation eines Solargenerators für eine hochfliegende Solarplattform* (Braunschweig, Technische Universität Braunschweig, 2021); Seite: 31
- [36] Diffusstrahlung; Wikipedia; Zugriff am 16.12.21; <https://de.wikipedia.org/wiki/Diffusstrahlung>; Seite: 31
- [37] Walter Fichter / Werner Grimm; Universität Stuttgart: *Script: Flugmechanik*; Shaker Verlag Aachen 2009; Seite: 35, 38, 39
- [38] Winfried Kassera *Motorflug Kompakt* (Stuttgart, Motorbuch Verlag, 8.Auflage 2020); Seite: 40
- [39] Fluggeschwindigkeit; Wikipedia; Zugriff am 13.12.21; <https://de.wikipedia.org/wiki/Fluggeschwindigkeit>; Seite: 40
- [40] Andreas Strohmayr; Universität Stuttgart: *Script: Luftfahrttechnik*; Rev. 1.03; Seite: 40, 42, 43, X, XXIII, XXIV
- [41] Rudolf Voit-Nietschmann; Universität Stuttgart: *Script: Flugzeugentwurf II; Begleitmaterial zu den Vorlesungen*; 2006; Seite: 42
- [42] Thorsten Lutz; Universität Stuttgart: *Skript zu den Vorlesungen Flugzeugaerodynamik I & II*; 2017; Seite: 43, 64
- [43] Cord-Christian Rossow; Klaus Wolf; Peter Horst: *Handbuch der Luftfahrzeugtechnik* (München, Carl Hanser Verlag, 2014); Seite: 40, 41, 43, 44, 46, 51
- [44] Andreas Strohmayr; Universität Stuttgart: *Script: Flugzeugentwurf II*; Rev. 1.02; Seite: 45
- [45] Daniel P. Raymer: *Aircraft Design: A Conceptual Approach*; 1985; Seite: 48
- [46] Aerodynamic Characteristics of Propellers; mh-aerotoools; Zugriff am 10.10.21; <https://www.mh-aerotoools.de/airfoils/index.htm>; Seite: 49, 50, 51
- [47] Helmut Lindner; Harry Brauer; Constans Lehmann: *Taschenbuch der Elektrotechnik und Elektronik* (Leipzig, Carl Hanser Verlag, 2008, 9.Auflage); Seite: 52, 53
- [48] Steffen Bazlen; Markus Blessing; Walter Commerell: *Anwendung von Batterieanalysemethoden zur Validierung von Alterungsmodellen in Lithium-Ionen Zellen* (Ulm, Technische Hochschule, 2020); Zugriff am 28.12.21; <https://www.cea-wismar.de/asimsst/data/043-ASIM-SST-2020-bazlen.pdf>; Seite: 57, 58
- [49] Agnes Klein: *Experimentelle Untersuchung von Batteriesystemen im simulierten niedrigen Erdorbit* (Stuttgart, Universität Stuttgart, 31.07.2017); Seite: 57, 58
- [50] Verifizierung; Wikipedia; Zugriff am 21.12.21; <https://de.wikipedia.org/wiki/Verifizierung>; Seite: 60
- [51] Modultest; Wikipedia; Zugriff am 22.12.21; <https://de.wikipedia.org/wiki/Modultest>; Seite: 60
- [52] Built-in test equipment; Wikipedia; Zugriff am 22.12.21; https://de.wikipedia.org/wiki/Built-in_test_equipment; Seite: 63
- [53] Hermann Friedrich; Frank Pietschmann: *Numerische Methoden* (Berlin, CPI Books GmbH, 2020, 2.Auflage); Seite: 66
- [54] High-Altitude Pseudo-Satellite (HAPS); Aurora Flight Sciences; Zugriff am 19.10.2021; <https://www.aurora.aero/odysseus-high-altitude-pseudo-satellite-haps/>; Seite: XII

- [55] Aurora Odysseus; Wikipedia; Zugriff am 19.10.2021;
https://en.wikipedia.org/wiki/Aurora_Odysseus; Seite: XII
- [56] Odysseus; the Solar-Powered Aircraft That Can Fly for Months; Popular Mechanics; Zugriff am 19.10.2021;
<https://www.popularmechanics.com/flight/a25101375/odysseus-solar-powered-aircraft/>; Seite: XII
- [57] Aurora's Odysseus – Large Enough for Its Mythic Name; Sustainable Skies; Zugriff am 19.10.2021; <http://sustainable skies.org/auroras-odysseus-large-enough-for-its-mythic-name/>;
Seite: XII
- [58] High-Altitude Pseudo-Satellites; AeroVironment; Zugriff am 19.10.2021;
<https://www.avinc.com/about/haps>; Seite: XIII
- [59] Pseudo-Satellit erreicht 19050 Meter; Flug Revue; Zugriff am 19.10.2021;
<https://www.flugrevue.de/zivil/aerovironment-sunglider-pseudo-satellit-erreicht-19050-meter/>;
Seite: XIII
- [60] Sun glider Builds on Legacy of Solar Aircraft; NASA; Zugriff am 19.10.2021;
<https://www.nasa.gov/centers/armstrong/features/sunglider-builds-on-legacy-of-solar-aircraft.html>; Seite: XIII
- [61] AeroVironment completes successful Sun glider HAPS stratospheric test flight; CompositesWorld; Zugriff am 19.10.2021;
<https://www.compositesworld.com/news/aerovironment-completes-successful-sunglider-haps-stratospheric-test-flight>; Seite: XIII
- [62] AeroVironment Helios; Wikipedia; Zugriff am 10.07.2021;
https://de.wikipedia.org/wiki/AeroVironment_Helios; Seite: XIII

A Verwendete Parameter und Daten

Anmerkung zu Symbolen: Die im Anhang verwendeten Symbole werden direkt neben ihrer Bezeichnung aufgeführt. Daher sind die hier verwendeten Symbole nicht im Symbolverzeichnis aufgenommen.

A.1 Flugzeugparameter HAP

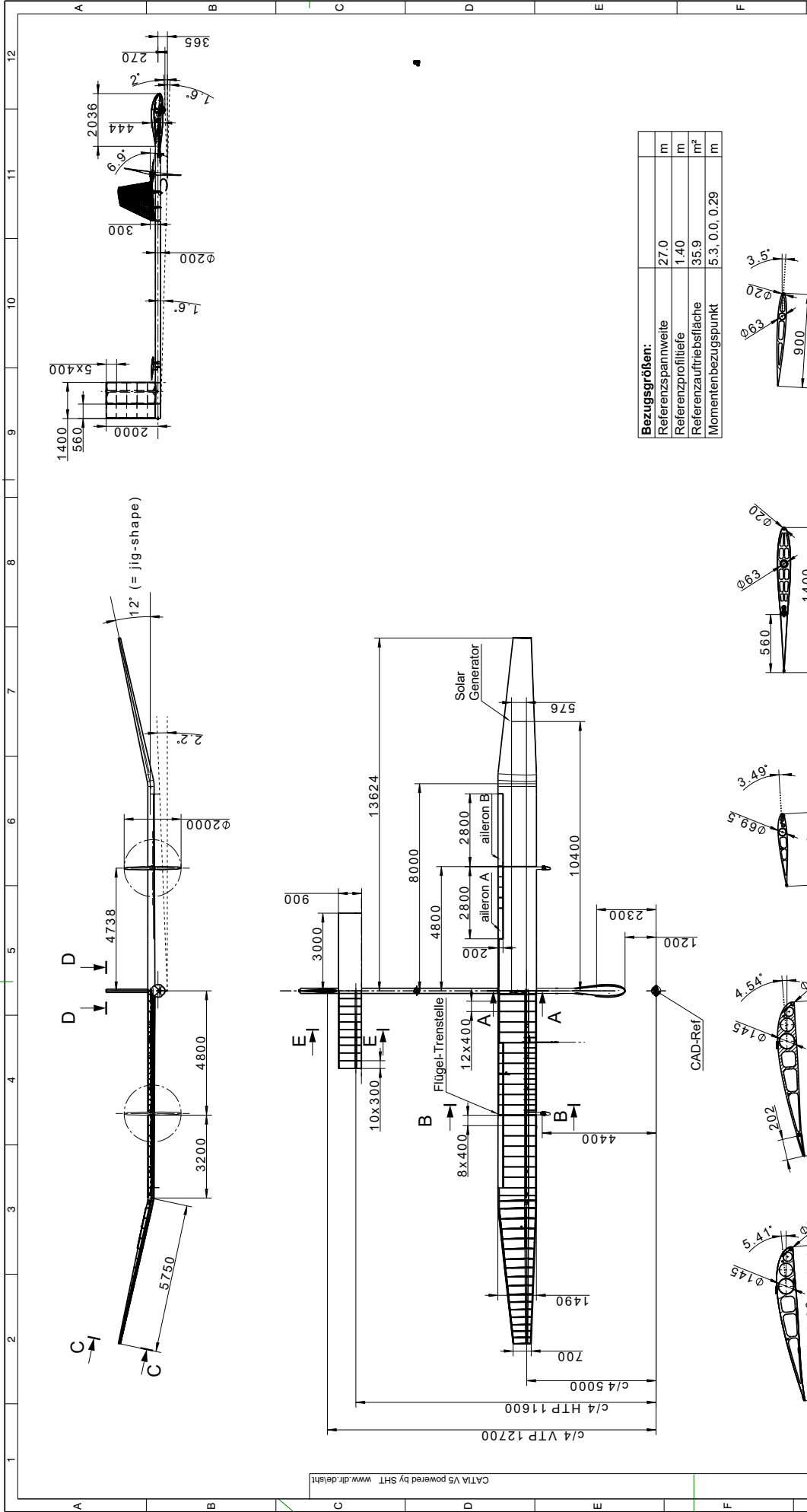
Im Folgenden sind die Eingabeparameter des Flugzeuges HAP aufgelistet. Diese wurden durch das DLR festgelegt. Felder mit der Kennzeichnung * wurden geschätzt, da diese Daten nicht vorhanden sind.

Parameter	Kürzel	Wert	Einheit	Bemerkung
Masse	m_{ac}	138	kg	
Flügelfläche	S_{ac}	35,9	m ²	
Schwerpunktslage	x_{CoG}	2,1	m	von Rumpfspitze
Flügel				
Flügelprofil		MH139R	-	
Flügelfläche	S	35,9	m ²	
Profiltiefe	c	1,4	m	
Streckung	Λ	21	-	
Oswaldfaktor	e	0,91	-	
Prop. factor	c_{D0fac}	1,25	-	wegen Solarpanele
Position $t1/4$	x_{ac}	2,0	m	von Rumpfspitze
Höhenleitwerk				
Flügelprofil		NACA0009	-	
Flügelfläche	S	4	m ²	
Profiltiefe	c	0,9	m	
Streckung	Λ	7	-	
Oswaldfaktor	e	0,8	-	
Prop. factor	c_{D0fac}	1,25	-	
Position $t1/4$	x_{ac}	12	m	von Rumpfspitze
Seitenleitwerk				
Nullwiderstand	c_{D0}	0,01	-	
Flügelfläche	S	2	m ²	
Profiltiefe	c	0,7	m	
Rumpf				
Nullwiderstand	c_{D0}	0,01	-	
Rumpffläche	S_{fus}	1	m ²	
Referenzlänge	c	12,5	m	

Tabelle A.1: Kenndaten des Flugzeuges HAP *1

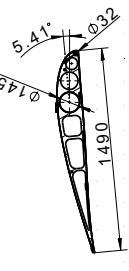
Parameter	Kürzel	Wert	Einheit	Bemerkung
Fahrwerk				
Nullwiderstand	c_{D0}	0,0057	-	
Fahrwerkfläche	S_{gear}	1	m ²	
Referenzlänge	c	1	m	
Antennen				
Nullwiderstand	c_{D0}	0,0037	-	
Antennenfläche	S_{fus}	1	m ²	
Referenzlänge	c	0,5	m	
Batterien				
Masse Batterien	m_{bat}	50	kg	
Energiedichte	e_{bat}	210	Wh/kg	
min. Ladestand	SoC_{min}	0,2	-	=:20 Prozent
Wirkungsgrad Laden	η_{chg}	0,98	-	
Wirkungsgrad Entladen	η_{dsg}	0,98	-	
Heiztemperatur	T_{heat}	10	°	
Solargenerator				
Fläche Solarzellen	A_{SG}	10	m ²	
Wirkungsgrad	η_{SG}	0,27	-	
Propeller				
Anzahl Propeller	N_{props}	2	-	
Propellerdaten	-	PROPMH2	-	
Motor				
Motordaten	-	MOTHAP	-	
max. Leistung des Motors	$P_{max\ Mot}$	2000	W	
max. Leistung Rekuperation	$P_{max\ rcu\ Mot}$	1000	W	
Drehmoment	M_{Mot}	+/-15	Nm	
Drehmomentkonstante	$k_M\ Mot$	0,2	1/Nms	
Leerlaufdrehzahl	$n_0\ Mot$	50	1/s	
elektr. Wirkungsgrad	$\eta_{el\ Mot}$	0,9	-	
Systeme				
Leistung	P_{sys}	100	W	

Tabelle A.2: Kenndaten des Flugzeuges HAP *2

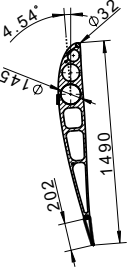


Bezugsgrößen:

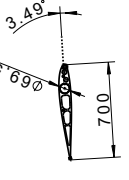
Referenzspannweite	27.0	m
Referenzprofiltiefe	1.40	m
Referenzaufltriebsfläche	35.9	m ²
Momentenbezugspunkt	5.3, 0.0, 0.29	m



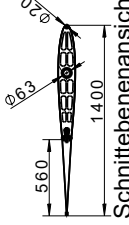
Schnittebenenansicht A-A
Maßstab: 1:25



Schnittebenenansicht B-B
Maßstab: 1:25



Schnittebenenansicht C-C
Maßstab: 1:25



Schnittebenenansicht D-D
Maßstab: 1:25

Schnittebenenansicht E-E
Maßstab: 1:25

Außenflügel-Rippenabstände	[m]	Außenflügel-Rippenabstände	[m]
Rippe 1 - 2	0.4	Rippe 8 - 9	0.35
Rippe 2 - 3	0.4	Rippe 9 - 10	0.35
Rippe 3 - 4	0.4	Rippe 10 - 11	0.35
Rippe 4 - 5	0.4	Rippe 11 - 12	0.3
Rippe 5 - 6	0.4	Rippe 12 - 13	0.3
Rippe 6 - 7	0.4	Rippe 13 - 14	0.3
Rippe 7 - 8	0.35	Rippe 14 - 15	0.3
		Rippe 15 - 16	0.25
		Rippe 16 - 17	0.25
		Rippe 17 - 18	0.25

1	05.10	Payload	SN	Name	Bewertung	Maßstab
					HAP-Omega-06	1:100
					HAP-Omega-06-Assembly	Genert
Zerlegungsnr.:						
006-000-00-001-WM						
ANDERENEN BING NUR ÜBER						
GAD-CATIA ZULASSIG						
30-Prüfung:						
14 MILLIMETER						
Format DIN A2						

Copyright © 2020 SHT. Alle Rechte vorbehalten. Nachdruck, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der SHT. CATIA V5 powered by SHT www.dlr.de/sht

Kennfelder NACA0009

In den Abbildungen A.1, A.2 und A.3 sind die Kennfelder des Pendelleitwerkes für das Flügelprofil NACA0009 gegeben.

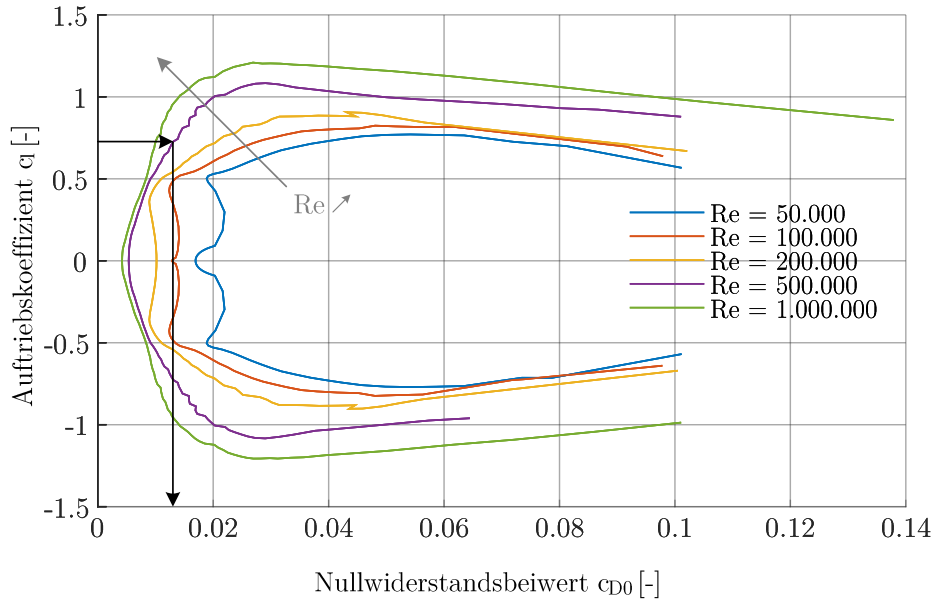


Abbildung A.1: Profilpolare des Pendelleitwerks für NACA0009 [1]

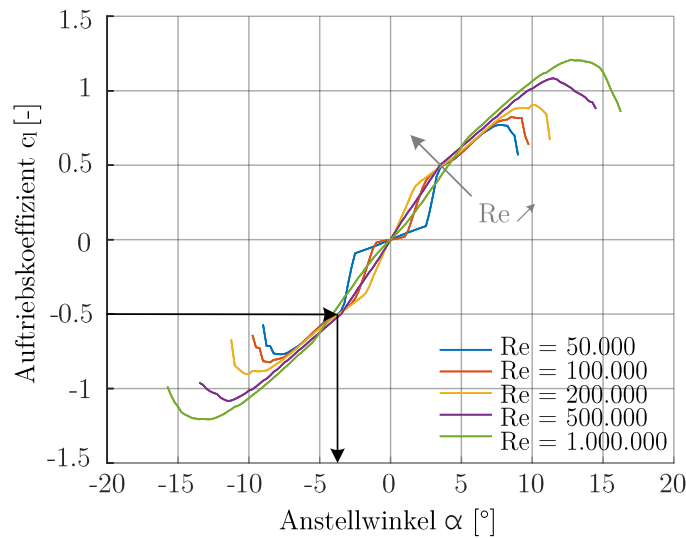


Abbildung A.2: Auftriebskoeffizient über Anstellwinkel für NACA0009 nach [1]

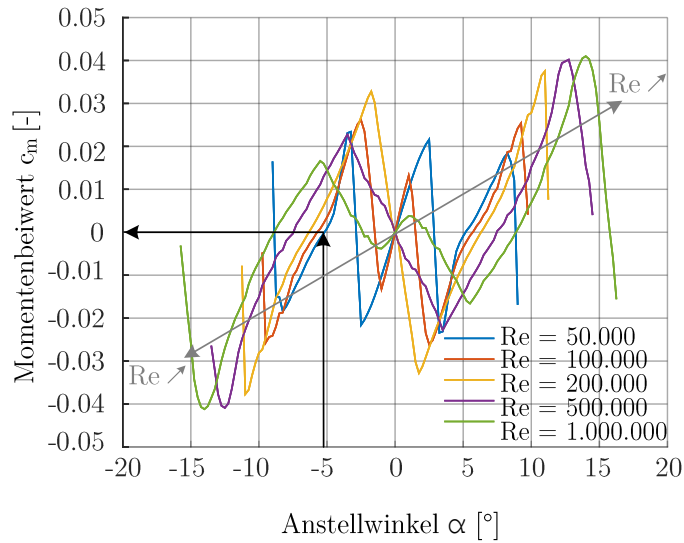


Abbildung A.3: Momentenbeiwert über Anstellwinkel für NACA0009 nach [1]

A.2 Umgebungsbezogene Daten

A.2.1 Sonnenmodell

Ergänzend zu den Abbildungen in **Kapitel 5** zur Verifikation des Sonnenmodells sind in Abbildung A.4 die Komponenten der Transformationsmatrix T_{ge} gezeigt.

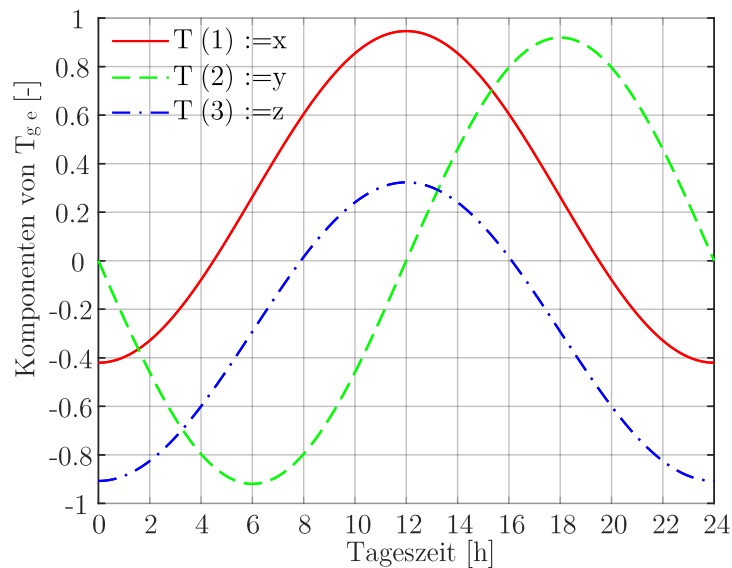


Abbildung A.4: Komponenten der Transformationsmatrix T_{ge} im Winter für $\phi = 48^\circ \text{ N}$; $h = 20.000 \text{ m}$

Die Abbildungen A.5 A.6 und A.7 zeigen die Sonnenverläufe im Sommer (=Tag 164) am Breitengrad 48° Nord in einer Flughöhe von 20.000 m . Sie decken sich mit den Sonnenmodellldaten aus [34].

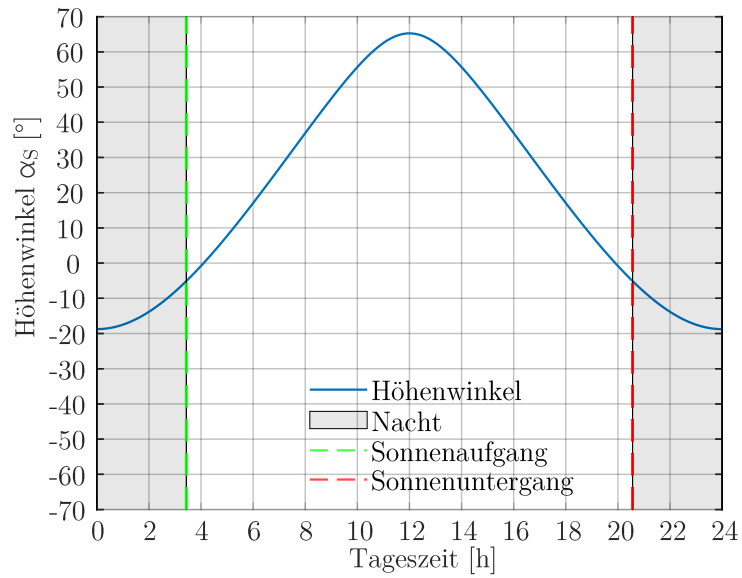


Abbildung A.5: Sonnenverlauf im Sommer für $\phi = 48^\circ \text{ N}$; $h = 20.000 \text{ m}$

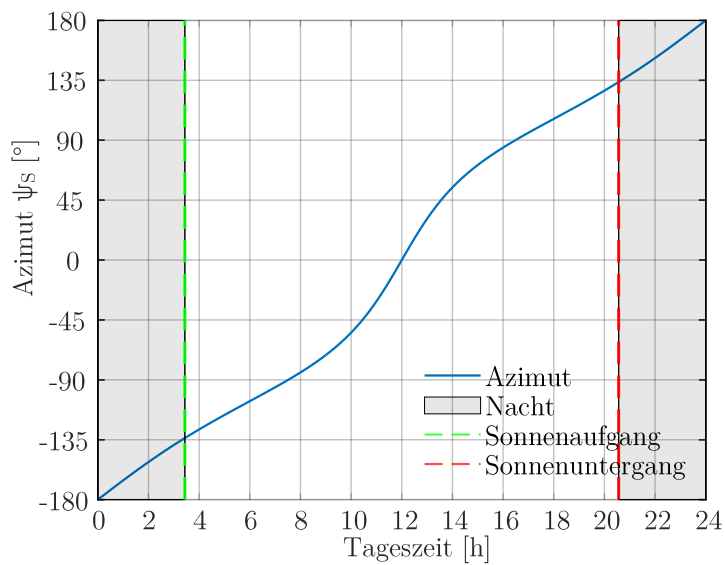


Abbildung A.6: Azimutverlauf im Sommer für $\phi = 48^\circ \text{ N}$; $h = 20.000 \text{ m}$

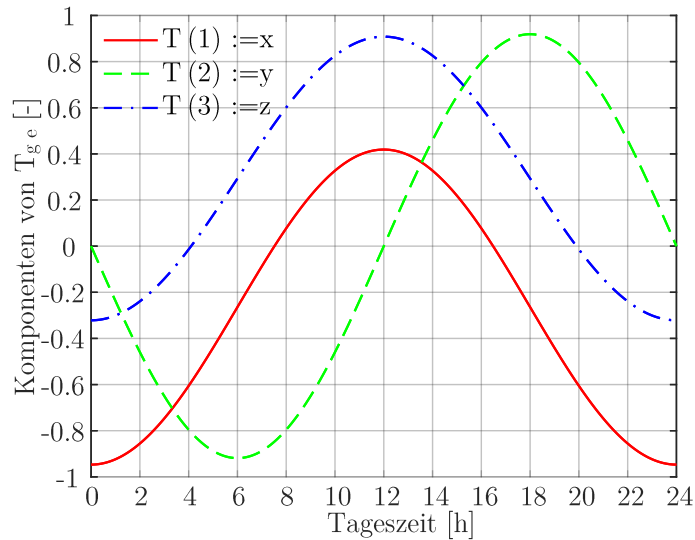


Abbildung A.7: Komponenten der Transformationsmatrix T_{ge} im Sommer für $\phi = 48^\circ \text{ N}$; $h = 20.000 \text{ m}$

A.2.2 Winddaten

Die Koordinaten der zu den Winddatensätzen gehörigen Positionen sind in Tabelle A.3 aufgeführt.

Position	Breitengrad	Längengrad
Hamburg	$53^\circ 33' 3,906'' \text{ N}$	$9^\circ 59' 37,255'' \text{ O}$
Kiruna	$67^\circ 51' 21'' \text{ N}$	$20^\circ 13' 31'' \text{ O}$
Yuma	$32^\circ 42' \text{ N}$	$114^\circ 37' \text{ W}$

Tabelle A.3: Geographische Breiten- und Längengrade der Windmodelle

A.3 Ergänzende Formeln

Die Flügelfläche eines zweigeteilten Flügels mit gerader und trapezförmiger Form ergibt sich mit A.1:

$$S_{wing} = 2 \cdot c_i \cdot l_{kink} + \frac{c_i + c_o}{2} \cdot (b - 2 \cdot l_{kink}) \quad (\text{A.1})$$

mit: S_{wing} : Flügelfläche [m^2]
 c_i : Tiefe der Flügelinnenseite [m]
 l_{kink} : Länge bis zum Knick des Flügels [m]
 c_o : Tiefe der Flügelaußenseite [m]
 b : Spannweite des Flügels [m]

A.4 Ergänzende Daten

A.4.1 BAE PHASA 35

Die Daten in Tabelle A.4 stammen aus den Quellen [3] [4] [5] [6] [7].

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{mtom}	150	kg	
Nutzlast	m_L	15	kg	
Spannweite	b_{wing}	35	m	
Fluggeschwindigkeit	v_{cruise}	93-145	km/h	Annahme TAS
Flughöhe	h_{cruise}	15-21	km	
max. Flughöhe	h_{max}	21336	m	
Geplante Einsatzdauer	$t_{mission}$	3	Jahre	
Generatorleistung	P_{el}	300-1000	W	
max. Breitengrad	ϕ	35	°	für dauerhafter Flug
Solarzellentyp	-	GaAs	-	
Batterietyp	-	Li-Ion	-	

Tabelle A.4: Kenndaten des Flugzeuges PHASA 35

Die Daten in Tabelle A.5 wurden mittels Ausmessen von [7] und der Annahme einer Spannweite b_{wing} von 35 m ermittelt.

Parameter	Kürzel	Wert	Einheit	Bemerkung
Spannweite	b_{wing}	35	m	
Flügelteufe innen	c_{wing_i}	0,8	m	
Flügelteufe außen	c_{wing_o}	0,6	m	
Flügelänge bis Knick	l_{kink}	8	m	von Mitte aus von Rumpfspitze
Position Flügel	x_{wing}	1,4	m	bis Flügelvorderkante
Rumpflänge	l_{fus}	14	m	
Spannweite HLW	b_{HTail}	5,2	m	
Tiefe HLW außen	c_{HTaili}	1,5	m	
Tiefe HLW innen	c_{HTailo}	0,8	m	
Tiefe SLW	c_{VTail}	1,4	m	gemittelt
Höhe SLW	h_{VTail}	2,9	m	
Propeller Durchmesser	D_{Prop}	2,5	m	

Tabelle A.5: Geometrische Daten zu PHASA 35

Mit den Daten aus Tabelle A.4 und A.5 können weitere Daten berechnet werden, die in Tabelle A.6 aufgelistet sind.

Parameter	Kürzel	Wert	Einheit	Bemerkung
Flügelfläche	S_{wing}	26	m^2	
Streckung Flügel	Λ_{wing}	22,9	-	
Höhenleitwerksfläche	S_{HTail}	5,98	m^2	
Streckung HLW	Λ_{HTail}	4,52	-	
Seitenleitwerksfläche	S_{VTail}	4,06	m^2	

Tabelle A.6: Berechnete Daten zu PHASA 35

A.4.2 Zephyr S

Die Daten in Tabelle A.7 stammen aus den Quellen [8] [10] [11].

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{mtom}	75	kg	
Nutzlast	m_L	5	kg	
Spannweite	b_{wing}	25	m	
Fluggeschwindigkeit	v_{cruise}	-	km/h	
Flughöhe	h_{cruise}	21	km	
max. Flughöhe	h_{max}	21195	m	
Batterietyp	-	Li-Si	-	

Tabelle A.7: Kenndaten des Flugzeuges Zephyr S

Die Daten in Tabelle A.8 wurden mittels Ausmessen von [12] und der Annahme einer Spannweite b_{wing} von 25 m ermittelt.

Parameter	Kürzel	Wert	Einheit	Bemerkung
Spannweite	b_{wing}	25	m	
Flügeltiefe innen	c_{wing_i}	1,5	m	
Flügeltiefe außen	c_{wing_o}	1,0	m	
Flügelänge bis Knick	l_{kink}	7	m	von Mitte aus
Position Flügel	x_{wing}	1,9	m	von Rumpfspitze bis Flügelvorderkante
Rumpflänge	l_{fus}	7,2	m	
Spannweite HLW	b_{HTail}	4,7	m	
Tiefe HLW außen	c_{HTaili}	0,7	m	
Tiefe HLW innen	c_{HTailo}	0,7	m	
Tiefe SLW	c_{VTail}	0,66	m	gemittelt
Höhe SLW	h_{VTail}	1,5	m	
Propeller Durchmesser	D_{Prop}	1,7	m	

Tabelle A.8: Geometrische Daten zu Zephyr S

Die Daten aus Tabelle A.9 wurden mit den Werten aus Tabelle A.7 und A.8 berechnet.

Parameter	Kürzel	Wert	Einheit	Bemerkung
Flügelfläche	S_{wing}	34,75	m^2	
Streckung Flügel	Λ_{wing}	17,98	-	
Höhenleitwerksfläche	S_{HTail}	3,29	m^2	
Streckung HLW	Λ_{HTail}	6,71	-	
Seitenleitwerksfläche	S_{VTail}	0,99	m^2	

Tabelle A.9: Berechnete Daten zu Zephyr S

A.4.3 Daten für Entwurfsdiagramm

Im Folgenden sind die Einzeldaten zu den Entwurfsdiagrammen in **Kapitel 2** aufgelistet.

Die Daten zu HAP sind aus [1]. Datenquellen zu BAE PHASA 35 aus [3] [4] [5] und Zephyr S aus [8] [10]. Datenquellen sind zu E Genius aus [13], AS33Me aus [14] [15] und Icare 2 aus [16] entnommen.

Die Flächenbelastung kann mit Formel A.2 nach [40] berechnet werden. Der Leistungsparameter wird mit Formel A.3 nach [40] berechnet.

$$\text{Flächenbelastung} = \frac{m_0}{S} \quad (\text{A.2})$$

$$\text{Leistungsparameter} = \frac{P}{m_0 \cdot g} \quad (\text{A.3})$$

mit: m_0 : Flugzeugmasse [kg]
 S : Flügelfläche [m^2]
 P : Leistung [W]
 g : Erdbeschleunigung [m/s^2]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	128	kg	
Flügelfläche	S_{wing}	36,03	m^2	
Antriebsleistung	P	2·4000	W	

Tabelle A.10: Kenndaten zu HAP Stand 15.05.21 [1]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	135	kg	
Flügelfläche	S_{wing}	26	m^2	
Antriebsleistung	P	2·4000	W	

Tabelle A.11: Kenndaten zu BAE PHASA 35 [3] [4] [5]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	80	kg	
Flügelfläche	S_{wing}	30	m^2	
Antriebsleistung	P	2·3000	W	

Tabelle A.12: Kenndaten zu Zephyr S [8] [10]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	915	kg	
Flügelfläche	S_{wing}	14,3	m^2	
Antriebsleistung	P	72000	W	

Tabelle A.13: Kenndaten zu EGenius [13]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	600	kg	
Flügelfläche	S_{wing}	10	m^2	
Antriebsleistung	P	35000	W	

Tabelle A.14: Kenndaten zu AS33Me [14] [15]

Parameter	Kürzel	Wert	Einheit	Bemerkung
Abflugmasse	m_{tow}	344	kg	
Flügelfläche	S_{wing}	25	m^2	
Antriebsleistung	P	12000	W	

Tabelle A.15: Kenndaten zu Icare II [16]

B HALE Flugzeuge

B.1 Aurora Flight Sciences Odysseus

Das Flugzeug Aurora Odysseus von der Firma Aurora Flight Sciences, einer Tochtergesellschaft der Firma Boeing, absolvierte seinen ersten Stratosphärenflug im Jahr 2019. Es wird mit dem Kürzel HAPS bezeichnet und steht für High-Altitude Pseudo Satellite. Damit wird verdeutlicht, dass Nutzlasten verwendet werden können, die sonst auf Satelliten im Einsatz sind. Die Spannweite beträgt 74 m die auf 3 Flugzeugrümpfe mit jeweils einem Pendelhöhen- und ein Seitenleitwerk verteilt sind. Eine weitere Besonderheit, neben den Solarzellen auf dem Flügel, ist der Einsatz von Solarpanelen auf den Rümpfen, sowie auf den Seitenleitwerken. Außerdem sind 6 Propeller im Einsatz, die ihre Energie entweder aus Lithium-Polymer Batterien oder aus den Gallium-Arsenide Solarzellen beziehen. Dadurch kann bei 25 kg Nutzlast eine Flughöhe von 20000 m oder bei 63,5 kg Nutzlast eine Flughöhe von 18288 m erreicht werden. Die Gesamtmasse wird auf 680 kg geschätzt. Die Einsatzdauer beträgt bei mittleren Breitengraden über ein Jahr hinaus und sogar bei arktischen Breitengraden noch 6 Monate [54] [55] [56] [57]. Abbildung B.1 zeigt das Flugzeug in Stratosphärenhöhe wie es seit 2019 geflogen wird.

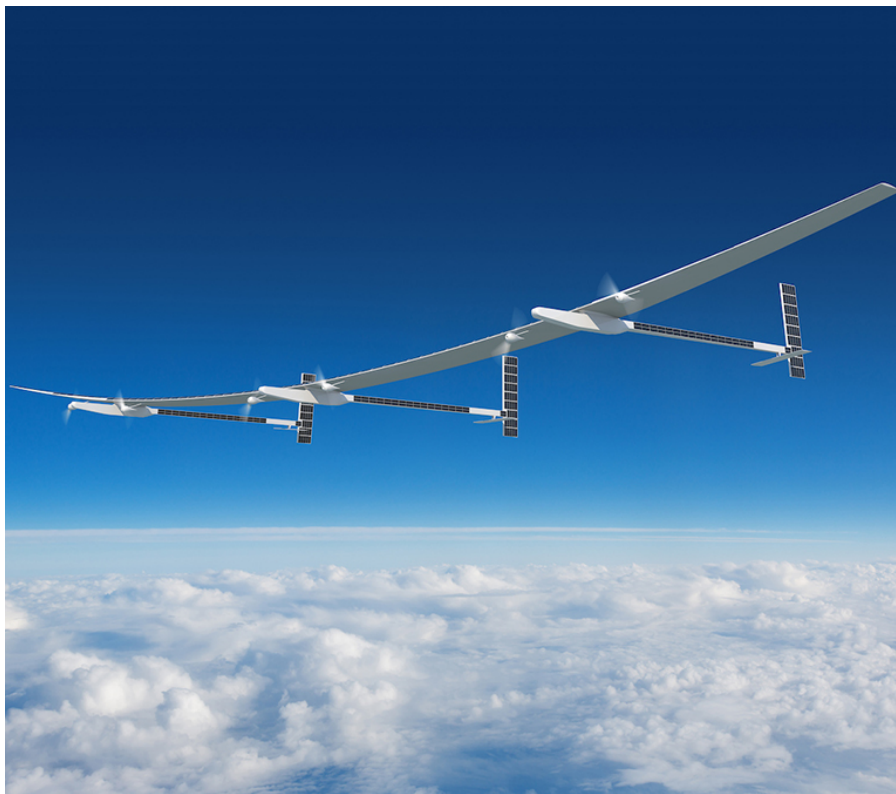


Abbildung B.1: Das Solarflugzeug Aurora Odysseus [54]

B.2 AeroVironment Sunlider

Der Sunlider der Firma AeroVironment baut auf eine Nurflügelkonfiguration auf. In Zusammenarbeit mit der NASA und weiteren beteiligten Firmen wurden bereits Vorgängermodelle entwickelt und erprobt. Aufzuzählen sind hier die Modelle Pathfinder, Centurion und Helios. Letzterer ist auf Abbildung B.2 gezeigt und besonders auffällig durch den Nurflügel.

Die aktuellste Version bildet der Sunlider mit einer Spannweite von 78 m und 10 verteilten Elektroantrieben mit Propellern. Dadurch wird eine Flughöhe von 19800 m bei einer Reisegeschwindigkeit

von 100 km/h erreicht. Die Nutzlastkapazitäten liegen bei bis zu 70 kg [58] [59] [60]. Abbildung B.3 zeigt die Aufnahme vom Flügel des Sungliders während eines Stratosphärenfluges im Jahr 2020.



Abbildung B.2: Das Flugzeug Helios von der NASA [62]



Abbildung B.3: Aufnahme des Sunglider in der Stratosphäre [61]

C Projektverlauf HAP

Das Projekt HAP wurde im Jahr 2019 mit dem System Specification Review (SSR) gestartet bei dem die Projektziele und Kenngrößen definiert wurden. Innerhalb des Projekts arbeiten mehrere Institute des DLR an verschiedenen Standorten gemeinsam an der Umsetzung. Das Preliminary Design Review (PDR) wurde Ende 2020 durchgeführt. Verschiedene Konzepte und Komponenten, sowie Fertigungsverfahren wurden bis dahin erstmalig untersucht und validiert. Das Critical Design Review (CDR) ist aktuell angesetzt für das erste Quartal 2022. Im Zuge des CDR stehen sämtliche Komponenten und Verfahren inklusive entsprechender Kennwerte fest. Danach erfolgt die Umsetzung des Erstdemonstrators HAP α . Nach erfolgreichem Erstflug Ende 2022 werden weitere Meilensteine zeitlich neu bewertet und festgelegt. Abbildung C.1 zeigt sämtliche Meilensteine und die Einordnung der Masterthesis.

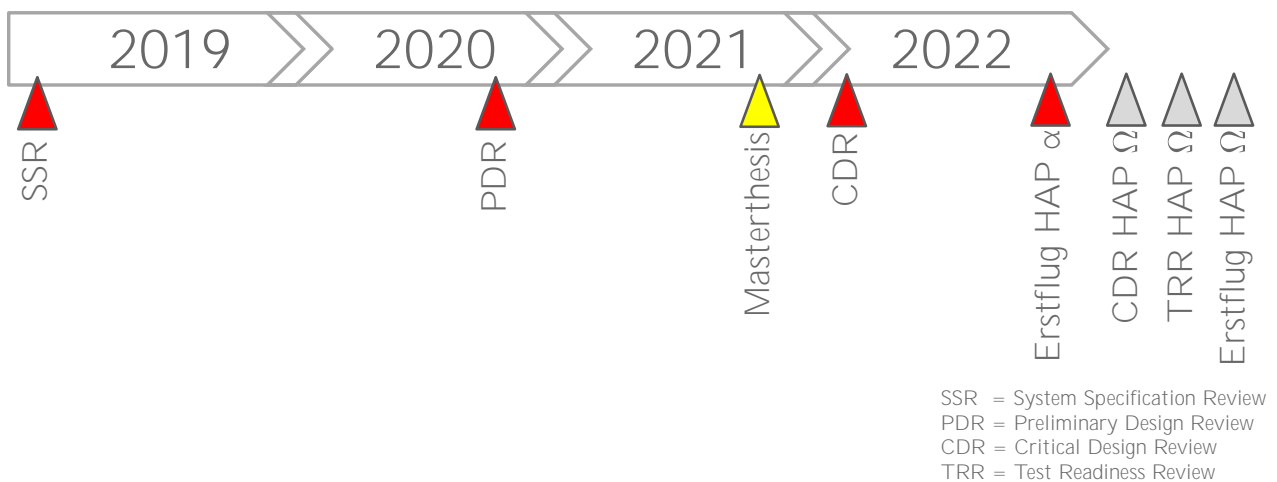


Abbildung C.1: Zeitlicher Verlauf Projekt HAP

Wie in Abbildung C.1 sichtbar ist nach HAP α ein weiterer Demonstrator gelistet und ist mit HAP Ω bezeichnet. Beide Flugzeuge sind geometrisch identisch und entsprechen der projektierten Dimension. Einzig die Fläche der applizierten Solarzellen und die Energiedichte der Batterien unterscheidet sie. Kennzahlen dazu sind in Tabelle C.1 gezeigt. Gründe für die Unterschiede sind primär die hohen Kosten für die Solarzellen und die Batterien. Ein Statosphärenflug über mehrere Tage ist für HAP α nicht geplant [1].

	HAP α	HAP Ω	
Energiedichte Batterie	250	350	<i>Wh/kg</i>
Fläche Solargenerator	11	22	<i>m²</i>

Tabelle C.1: Unterschiede HAP α und HAP Ω

D Programmarchiv

Anmerkung zu Symbolen: Die im Anhang verwendeten Symbole werden direkt neben ihrer Bezeichnung und/oder Formel aufgeführt. Daher sind die hier verwendeten Symbole nicht im Symbolverzeichnis aufgenommen.

Es werden im Folgenden Zusatzinformationen zum Programm und/oder Analysen sowie ältere oder nicht mehr genutzte Programmstände und -funktionen aufgeführt.

D.1 Parameterliste des Programms

Anmerkung zum Stand: Die Parameterliste wurde zum Ende der Arbeit verfasst mit dem Parameterstand vom 30.12.2021. Daher sind Abweichungen zum Stand in den Kapiteln möglich. Temporäre Parameter innerhalb der Funktionen sind nicht gelistet.

Parameter	Einheit	Bedeutung
Klasse Scenario		
Aircraft		Objekt
Environment		Objekt
Mission		Objekt
SimLog		Objekt
time	s	Aktuelle Zeit
day	-	Aktueller Tag
iter		Feld mit Iterationsparametern
x_g	m	X-Koordinate im geodätischen KOS
y_g	m	Y-Koordinate im geodätischen KOS
z_g	m	Z-Koordinate im geodätischen KOS
alt	m	Aktuelle Höhe
<i>iter.*</i>		
.mode		Modus der Simulation test1, testn, full
.dt	s	Zeitschritt für Simulation
.ds	m/s	Schrittweite zur Lastreduktion von Geschwindigkeiten
.da	rad	Schrittweite zur Lastreduktion von Winkeln
.sim_round	-	individueller Zähler
.sim_maxrounds	-	max. Anzahl an Simulationsrunden
.threshold	-	Schwellwert für Iterationen
.counter_max	-	max. Anzahl an Iterationen
.checkpoint	min	Zeitmarkierung in Flugpfadausgabe
.delta_0	-	Startdifferenz
.counter_0	-	Startzähler
.overload	-	Überlast Indikator
Klasse Aircraft		
v_min_sl	m/s	min. Fluggeschwindigkeit auf SL
v_min_FL	m/s	min. Fluggeschwindigkeit TAS in Flughöhe
v_max_sl	m/s	max. Fluggeschwindigkeit auf SL

v_max_FL	m/s	max. Fluggeschwindigkeit TAS in Flughöhe
max_op_FL	m	max. Betriebsflughöhe
m	kg	Gesamtgewicht Flugzeug
S	m^2	Flügelfläche
x_CoG	m	Schwerpunktlage bezogen auf Flugzeugnase
Wing		Objekt (Typ Winglike)
Htail		Objekt (Typ Winglike)
Vtail		Objekt (Typ DragProducer)
Fuselage		Objekt (Typ DragProducer)
LandingGear		Objekt (Typ DragProducer)
Antennas		Objekt (Typ DragProducer)
Battery		Objekt
SolarGenerator		Objekt
Propulsion		Objekt
P_systems	W	elektr. Leistung System
alpha_Wing	rad	Anstellwinkel Flügel
alpha_Tail	rad	Anstellwinkel Höhenleitwerk
Re_Wing	-	Reynoldszahl Flügel
CL	-	Auftriebskoeffizient
CD	-	Widerstandskoeffizient
v	m/s	Fluggeschwindigkeit
v_h	m/s	Fluggeschwindigkeit horizontal proj.
v_v	m/s	Fluggeschwindigkeit vertikal proj.
r_turn	m	Kurvenradius des geflogenen Kreises
theta	rad	Nickwinkel
gamma	rad	Bahnneigungswinkel
phi_roll	rad	Rollwinkel
psi	rad	Gierwinkel
N_props	-	Anzahl der Propeller
Opt		Feld mit Optimalparametern
Iterlog		Feld mit Iterationsmarkern
<hr/> <i>Iterlog.*</i>		
.V	s	Marker für Geschwindigkeit
.CL	s	Marker für Auftriebskoeffizient
.thrust	s	Marker für Schub
.motor	s	Marker für Motor
.battery	s	Marker für Battery
<hr/> <i>Opt.*</i>		
.CL_eps_min	-	Auftriebskoeffizient für bestes Gleiten
.CL_gamma_max	-	Auftriebskoeffizient für bestes Steigen
<hr/> Klasse Mission		
timestamp_0	s	Startuhrzeit im 24h System
daystamp_0	-	Starttag bezogen auf 365 Tage im Jahr
phi	rad	Breitengrad
lambda	rad	Längengrad
section		Missionsabschnitt

section_type		Art des Missionsabschnittes
S_names		Feld mit Abschnittsnamen
S_data		Feld mit Abschnittsdaten
FL_initial	m	Startflughöhe
FL_target	m	Zielflughöhe
delta	-	Differenzparameter
 <i>S_data.*</i>		
.(1,i)	rad	Bahnneigungswinkel
.(2,i)	m	Radius des zu fliegenden Flugkreises
.(3,i)	m/s	Fluggeschwindigkeit in EAS
.(4,i)	m	Zielflughöhe
.(5,i)	h	Zeit in Segment
.(6,i)		Priorität
		FL, time, clocktime
 Klasse		
Environment		
<hr/>		
Atmosphere		Objekt
Sun		Objekt
Wind		Objekt
 Klasse SimLog		
<hr/>		
DataNames	-	Feld mit Datennamen
Data	-	Feld mit Datenwerten
n_data	-	Anzahl der zu speichernden Daten
timestamp_0	s	Zeitstempel zu Simulationsbeginn
 Klasse		
Winglike		
<hr/>		
Name		Name des Flügelprofils
Data		Kennfelder
Re	-	Reynoldszahl
alpha	rad	Anstellwinkel
S	m^2	Flügelfläche
S_wet	m^2	Bespülte Oberfläche
c	m	Flügeltiefe
lambda	-	Streckung
eff	-	Oswaldfaktor
CD0_fac	-	Aufschlagsfaktor wegen Solarpanele
x_AC	m	Position aerodynamische Zentrum bezogen auf Flugzeugnase
 Klasse		
DragProducer		
<hr/>		
CL	-	Auftriebskoeffizient
CD	-	Widerkoeffizient
S	m^2	Bespülte Oberfläche
lref	m	Referenzlänge

Klasse Battery

m	kg	Masse der Batterien
e	Ws/kg	Energiedichte
e_0	Ws/kg	Theor. Energiedichte
r_eTT25	-	Temperaturabhängigkeit Energiedichten
E	Ws	Eigentliche elektr. Energie
E_max	Ws	Max. elektr. Energie
E_charge	Ws	Geladene elektr. Energie
E_discharge	Ws	Entladene elektr. Energie
E_heat	Ws	Heizenergie
SoC	-	Ladestand der Batterien
SoC_min	-	Minimaler Ladestand der Batterien
eta_chg	-	Wirkungsgrad Laden
eta_dsg	-	Wirkungsgrad Entladen
zeta_dgr	-	Degradierungsfaktor
zeta_dgr_0	-	Degradierungsfaktor bei Start
zeta_dgr_n	-	Degradierungsfaktor nach Zyklen
n_chg	-	Äquivalenter Ladezyklus
T_heat	K	Zu heizende Batterietemperatur
HeatMap		Kennfeld zur Heizleistung

Klasse SolarGenerator

SolarPanels		Feld mit Daten zu Solarpanelen
N_Panels	-	Anzahl der Solarpaneele
A_SG	m^2	Fläche des Solargenerators
Eff_SG	-	Wirkungsgrad des Solargenerators
P_SG	W	Energie durch Solargenerator
p_SG	W/m^2	Spez. Energie durch Solargenerator
T_g_p		Transformationsmatrix geodätisch nach panel

Klasse Propulsion

Motor		Objekt
Propeller		Objekt
P_el_tot	W	Elektrische Leistung
N_Props	-	Anzahl der Propeller
Iterlog		Feld mit Iterationsmarkern

Klasse Propeller

D_p	m	Propellerdurchmesser
J	-	Kennfeld Fortschrittsgrad
CT	-	Kennfeld Schubkoeffizient
CP	-	Kennfeld Leistungskoeffizient
T_prop	N	Schub Propeller (gesamt)
P_prop_aero	W	Aerodyn. Leistung Propeller

Eta_prop	-	Wirkungsgrad Propeller
N_prop	1/s	Drehzahl Propellerwelle
P_prop	W	Leistung Propellerwelle
M_prop	Nm	Drehmoment Propellerwelle
T_min	N	Minimaler Schub
T_max	N	Maximaler Schub
<hr/>		
Klasse Motor		
<hr/>		
Parameters		Motor肯ndaten
Eta_el	-	Elektrische Wirkungsgrad
P_el	W	Elektrische Leistung
P_shaft	W	Leistung der Abtriebswelle
M_shaft	Nm	Drehmoment der Abtriebswelle
N_shaft	1/s	Drehzahl der Abtriebswelle
<hr/>		
Klasse Atmosphäre		
<hr/>		
Atmo	kg/m^3	Kennfeld mit Atmosphärenndaten
Rho_SL	μPas	Dichte auf SL
Eta_SL	-	Dynamische Viskosität auf SL
Nu_SL	m^2/s	Kinematische Viskosität auf SL
T	K	Temperatur
P	Pa	Druck
Rho	kg/m^3	Dichte
Nu	m^2/s	Kinematische Viskosität
A	m/s	Schallgeschwindigkeit
alt_env	m	Flughöhe Zwischenspeicher
<hr/>		
Klasse Sun		
<hr/>		
Eps_0	W/m^2	Solarkonstante
Eps_dir	W/m^2	Direkte solare Strahlung
Zeta_S	rad	Sonneneinfallswinkel
Alpha_S	rad	Höhenwinkel
Psi_S	rad	Azimut der Sonne
Delta	rad	Deklination der Sonne
Tau	rad	Stundenwinkel
Zeta_dep	rad	Depression
R_earth	m	Erdradius
T_g_e		Transformationsmatrix erdfest nach geodätisch
<hr/>		
Klasse Wind		
<hr/>		
Windmap		Feld mit Winddaten
v_w	m/s	Windgeschwindigkeit
psi_w	rad	Windrichtung
<hr/>		
Klasse Study		
<hr/>		
Scenario		Objekt
index	-	Interner Zähler

inputs		Feld mit Eingabeauswahl
max_rounds	-	Anzahl der Studienrunden
mode		Art der Studie
		single, double
param		Feld mit Parameterdaten für Studie
paths		Feld mit Pfadangaben der Studienparameter
progress	-	Interne Studienstandsanzeige
state		Status der Studie
StudyLog		Feld mit gesicherten Simulationsdaten
timer	s	Feld für Zeiten zu speichern

D.2 Analyse Stationierbarkeit über Nacht

Im Folgenden soll die Stationierbarkeit des Flugzeuges über Nacht analysiert werden bei Annahme einer Ladekapazität wie bei Missionsstart. Die verwendeten Parameter entsprechen denen der für die Standardmission verwendeten (siehe Anhang A). Es wird ein Reiseflug von 12 h bei einer Startzeit von 18 Uhr simuliert.

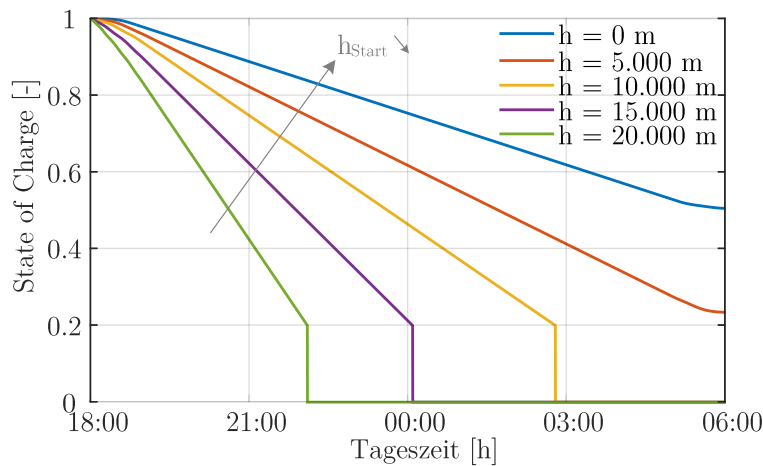


Abbildung D.1: Verlauf des SoC im Reiseflug über Nacht für verschiedene Flughöhen

In Abbildung D.1 sind die Ergebnisse des SoC dargestellt, welche ergeben, dass der SoC bei Missionsende für steigende Flughöhen abnimmt. Eine durchgehende Stationierung ab ca. 7.500 m und steigend ist laut der Abbildung nicht mehr gegeben. Dies resultiert aus der Erhöhung des Widerstandes durch die Abnahme der Reynoldszahl in höheren Flughöhen. Der erhöhte Widerstand führt zu entsprechend höherem Schub und wiederum zur schnelleren Entladung. Andere Einflüsse durch Faktoren wie beispielsweise der Propellerwirkungsgrad sind im Vergleich zur Erhöhung des Widerstandes vernachlässigbar klein.

D.3 Analyse bestes Gleiten

Es wird der Unterschied zum SoC am Missionsende der Standardmission bei Verwendung der Geschwindigkeit für bestes Gleiten ε_{min} untersucht. Dabei wird die Gesamtmissionsdauer an die Standardmission angepasst um vergleichbare Werte des SoC zu erhalten. Die Standardmission wird mit einer Fluggeschwindigkeit von 9 m/s EAS durchgeführt. Abbildung D.2 zeigt den Höhenverlauf beider Simulationen in welcher die Standardmission eine bessere Steigleistung aufzeigt.

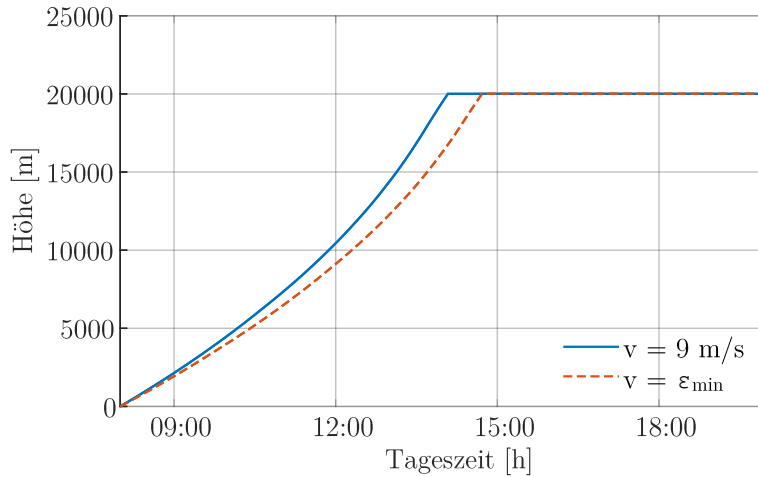


Abbildung D.2: Höhenverlauf für verschiedene Fluggeschwindigkeiten

Der SoC ist in Abbildung D.3 dargestellt und ergibt, dass die Verwendung der Fluggeschwindigkeit für bestes Gleiten eine Verbesserung um ca. 7 % bewirkt. Jedoch ist anzumerken, dass die gemittelte Fluggeschwindigkeit in EAS dabei einen Wert von $6,72 \text{ m/s}$ aufweist, welcher außerhalb der Flugbereichsgrenzen liegt.

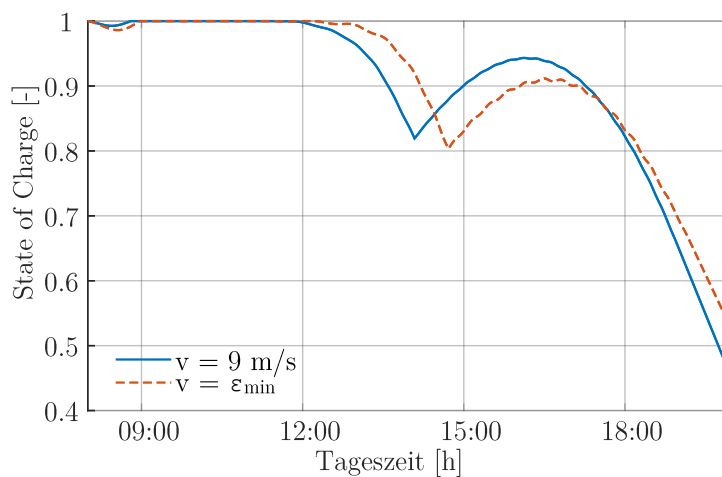


Abbildung D.3: Verlauf des SoC für verschiedene Fluggeschwindigkeiten

Eine Kombination aus bestem Steigen für den Steigflug und bestem Gleiten für den Reiseflug wird untersucht. Das Ergebnis ist in Abbildung D.4 dargestellt. Dabei wird eine Verbesserung des SoC um 14 % erzielt, weshalb ein Review der Flugbereichsgrenzen zur Maximierung des SoC empfohlen wird.

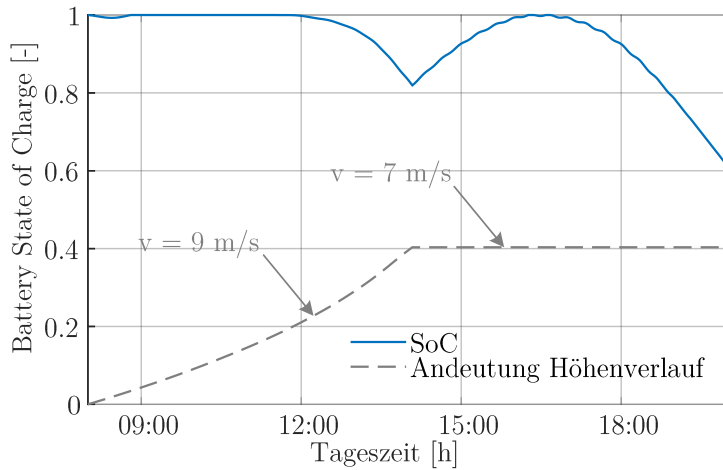


Abbildung D.4: Verlauf des SoC für beste Leistungsparameter

D.4 Analyse Sinkflug über Nacht

Nachfolgend wird untersucht, ob eine Stationierung über Nacht durch den Einsatz eines Sinkfluges realisiert werden kann. Die verwendeten Parameter entsprechen denen der für die Standardmission verwendeten (siehe Anhang A). Es wird ein Sinkflug von 15 h bei einer Startzeit von 18 Uhr simuliert. Darauf folgt ein Steigflug auf Reishöhe.

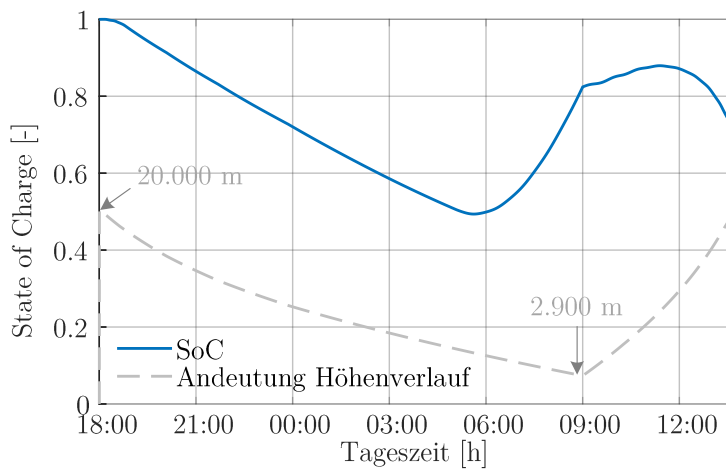


Abbildung D.5: Verlauf des SoC bei Sinkflug über Nacht

In Abbildung D.5 ist der Verlauf des SoC dargestellt. Trotz des Sinkfluges entlädt sich die Batterie bis ca. 50 %. Dies resultiert aus den Energieverbräuchern bestehend aus den Systemen, der Batterieheizung und des Leerlaufstroms der Antriebe. Abbildung D.6 zeigt die Ent- und Beladeenergien über der Missionszeit. Daraus wird ersichtlich, dass ab ca. 6 Uhr die Batterien wieder geladen werden, wodurch der Sinkflug reduziert werden könnte. Eine Verkürzung des Sinkfluges auf 6 Uhr ist dennoch nicht möglich, da der anschließende Steigflug den Batterieladestand unter sein Minimum entlädt.

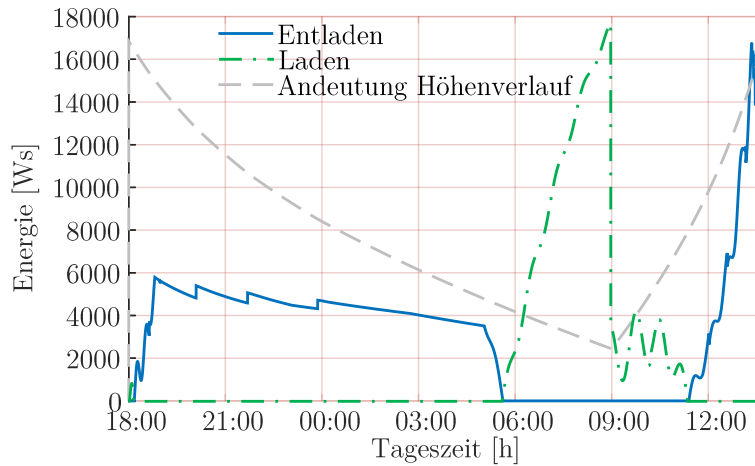


Abbildung D.6: Verlauf der Energien bei Sinkflug über Nacht

D.5 Berechnung der Luftdichte

Anmerkung: Diese Berechnung wird vom Programm nicht mehr aktiv genutzt.

Die Luftdichte ρ kann mit Formel D.1 aus [40] für die Troposphäre berechnet werden. Diese erstreckt sich bis 11.000 m Höhe. Für größere Höhen kann die Dichte mit Formel D.2 nach [40] berechnet werden. Der Höhenbereich von 11.000 m bis 50 km wird als Stratosphäre bezeichnet. Die Ergebnisse beider Berechnungsmethoden sind in Abbildung D.7 gezeigt.

$$\frac{\rho}{\rho_0} = \left(1 - \frac{n-1}{n} \cdot \frac{g}{R \cdot T_0} \cdot (H - H_0)\right)^{\frac{1}{n-1}} \tag{D.1}$$

$$\frac{\rho}{\rho_i} = e^{-\frac{g}{R \cdot T_i} \cdot (H - H_i)} \tag{D.2}$$

mit:	ρ	: Luftdichte	$[kg/m^3]$
	$\rho_0 = 1,225$: Luftdichte auf Meereshöhe	$[kg/m^3]$
	$n = 1,235$: Polytropenexponent	$[-]$
	$g = 9,81$: Erdbeschleunigung	$[m/s^2]$
	$R = 287,05$: Gaskonstante	$[J/(kgK)]$
	$T_0 = 288,15$: Temperatur auf Meereshöhe	$[K]$
	H	: Höhe	$[m]$
	$H_0 = 0$: Referenzhöhe	$[m]$
	$\rho_i = 0,36392$: Luftdichte nach Normatmosphäre	$[kg/m^3]$
	$T_i = 216,65$: Temperatur nach Normatmosphäre	$[K]$
	$H_i = 11.000$: Höhe nach Normatmosphäre	$[m]$

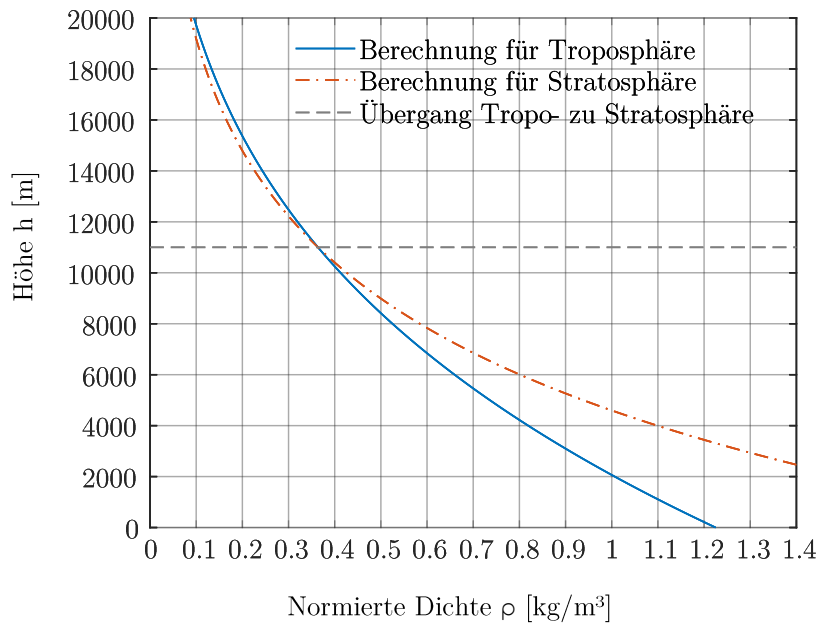


Abbildung D.7: Berechnung der Luftdichte nach [40]

Eidesstattliche Erklärung

Hiermit räume ich, Müller, Thomas, 3434440, 70569 Stuttgart, der Universität Stuttgart, Institut für Flugzeugbau, ein kostenloses, zeitlich und räumlich unbeschränktes, einfaches Nutzungsrecht an der von mir erstellten Bachelor,- Master – oder sonstigen studentischen Arbeit mit dem Titel

Entwicklung einer Missionssimulation für solarelektrische stationierbare Stratosphärenflugzeuge zur Analyse verschiedener Flugzeug-Konfigurationen

Ich erkläre, die Arbeit selbständig verfasst und bei der Erstellung dieser Arbeit die einschlägigen Bestimmungen, insbesondere zum Urheberrechtsschutz fremder Beiträge, eingehalten zu haben. Soweit meine Arbeit fremde Beiträge (z.B. Bilder, Zeichnungen, Textpassagen) enthält, erkläre ich, dass diese Beiträge als solche gekennzeichnet sind (z.B. Zitat, Quellenangabe) und ich eventuell erforderlich gewordene Zustimmungen der Urheber zur Nutzung dieser Beiträge in meiner Arbeit eingeholt habe. Für den Fall der Verletzung Rechte Dritter durch meine Arbeit, erkläre ich mich bereit, der Universität Stuttgart einen daraus entstehenden Schaden zu ersetzen bzw. die Universität Stuttgart auf deren Aufforderung von eventuellen Ansprüchen Dritter freizustellen. Das der Universität Stuttgart hiermit eingeräumte Nutzungsrecht erstreckt sich auf sämtliche bekannte Nutzungsarten und umfasst neben dem Recht auf Nutzung der Arbeitsergebnisse in Forschung, Lehre und Studium, insbesondere das Recht der Vervielfältigung und Verbreitung, das Recht zur Bearbeitung und Änderung inklusive Nutzung, Vervielfältigung und Verbreitung der dabei entstehenden Ergebnisse, sowie insbesondere das Recht der öffentlichen Zugänglichmachung im Internet sowie das Recht der Weiterübertragung auf einen Dritten ohne meine erneute Zustimmung. Mir ist bekannt, dass die Einräumung des Nutzungsrechts der öffentlichen Zugänglichmachung auch beinhaltet, dass mein Name im Zusammenhang mit dem Titel der oben genannten Arbeit auf den Webseiten der Universität Stuttgart genannt werden kann. Mir ist auch bekannt, dass – sofern meine Arbeit selbst nicht im Internet zugänglich gemacht wird – die Einräumung des Nutzungsrechts der öffentlichen Zugänglichmachung auch umfasst, dass die Universität Stuttgart auf ihren Webseiten meinen Namen im Zusammenhang mit dem Titel der oben genannten Arbeit (z.B. in Listen über am Institut abgeschlossene studentische Arbeiten) nennen kann. Außerdem übertrage ich der Universität Stuttgart das Eigentum an einem von mir der Bibliothek des Instituts für Flugzeugbau kostenlos zur Verfügung gestellten Exemplars meiner oben genannten Arbeit und räume der Universität Stuttgart auch für dieses Exemplar die oben genannten Nutzungsrechte ein.

Ort, Datum, Unterschrift